UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior – Leganés

INGENIERÍA INDUSTRIAL ELECTRÓNICA Y AUTOMÁTICA

TRABAJO FIN DE GRADO

DESING AND IMPLEMENTATION OF CONTROL HARDWARE

AUTHOR: Rodrigo Barbosa Moreno

TUTOR: Ramón Barber Castaño

DIRECTOR: Ramón Barber Castaño

Leganés, 2016

TITLE: DESING AND IMPLEMENTATION OF CONTROL HARDWARE

AUTHOR: Rodrigo Barbosa Moreno

DIRECTOR: Ramón Barber Castaño

TUTOR: Ramón Barber Castaño

The lecture of this End of the Grade Project was carried out in May the fourth of 2016

Graded by the following tribunal:

PRESIDENT: Marta Ruiz LLata

SECRETARY: Ricardo Valverde Gil

CHAIR: Alberto Emanuel Quintero Gámez

Obtaining the next rate:

RATING:

President **Secretary** **Chair**

[Cuando seas yunque, aguanta. Cuando seas martillo, golpea.]

ABSTRACT

The idea of adapting old mechatronic systems to new technologies leads to the creation of generic control hardware that upgrades them. With this in mind control hardware has been developed from the initial stages of design to the experimental tests. Altium software has been used to design the schematics and build the system conditioning circuit.

To obtain experimental results some Simulink blocks diagrams have been implemented using Matlab. The hardware designed has been tested in two laboratory models.

In addition, a study of the ESCs and the viability to use them as control systems for engines used in the automatic industry.

GENERAL INDEX

GENERAL INDEX

FIGURE INDEX

FIGURE INDEX

TABLE INDEX

1. Introduction

Automatic control systems have been experiencing a revolution along the past ten years, when the first low cost microcontroller boards emerged the market changed. Suddenly a new low cost gamut of products was design to fulfill every need that these boards could have, from sensors to drivers and even power sources. These new products had a good performance and quality but the most important feature was that they were built for general purpose which allow them to became the main pillar to continue the evolution of the automatic control systems. The drastic and almost not compatible change in the market left a lot of hardware and software outdated. Most part of these control systems remains unchanged because they were expensive and they still work or because there is no replacement for the hardware or the software, which leaves a lot of systems at risk.

The idea of adapting the old systems to the new technology was really challenging and once investigated the objective spread into reducing the size of the control systems and making it portable and variable.

1.1 Objetives

The main objective of this project is to design a control hardware for mechatronic systems. This idea came from the intention to upgrade some obsoletes systems. They were really good ones but old, big and were usually using serial ports to establish connection with a PC. Along with the drone

trend, the idea of using an ESC [4] to control the motors giving to the system more flexibility and reducing its size came across.

The perfect systems to develop this project were found in the laboratory classrooms, two completely different models which use old hardware to work. Taking all of this into account the main goals for this project are:

- Design a generic control system for the laboratory models using a microcontroller board.
- Implement a motor driver using an ESC and a microcontroller board.

1.2 Document structure

The content of this document is distributed in seven chapters. It starts with a description of all the hardware used, in Chapter 2. Chapter 3 describes the process followed to design and manufacture a PCB using Altium. Chapter 4explains what is an ESC, what can it do and how to control it. It is in Chapter 5 where all the tests and all the results obtained with this prototype are explained and documented. Chapter 6 details the socioeconomic impact of the project and a budget study. Finally all the conclusions and the future lines of work are shown in chapter 7.

2. Hardware

In this section the hardware used to develop this project is going to be briefly described in order to give a wider perspective of the elements involved in the project and its characteristics. The elements included in this section are:

- The STM32F4Discovery trainer board.
- Two laboratory models: pneumatic system and motor.
- The Motor Maxtom.

2.1 STM32F4Discovery board

The STM32F4Discoveryt [6] is a low cost trainer board, built by ST Microelectronics to evaluate ARM STM32 Cortex-M4 microcontrollers. It is the most important piece of the project since it will be used in to link the hardware with the software allowing the control of the different systems. As it is shown in Figure 1 the board can be connected directly to a computer by a mini USB port, it has 6 controllable LEDs, two buttons, integrated accelerometer, digital microphone and soldered in the middle the STM32F407VG microcontroller. This ARM Cortex-M4 microcontroller has an LQFP100 package and has to be connected to a single voltage source between 2 and 3.6 V which is described in Table 1, along with some other general features.

To program this microcontroller ST Microelectronics has developed ST-LINK which is an external tool used to program and debug. This system is embedded on the board and it allows both send one single

stream of source code to program the microcontroller or establish a link between the microcontroller and the pc.

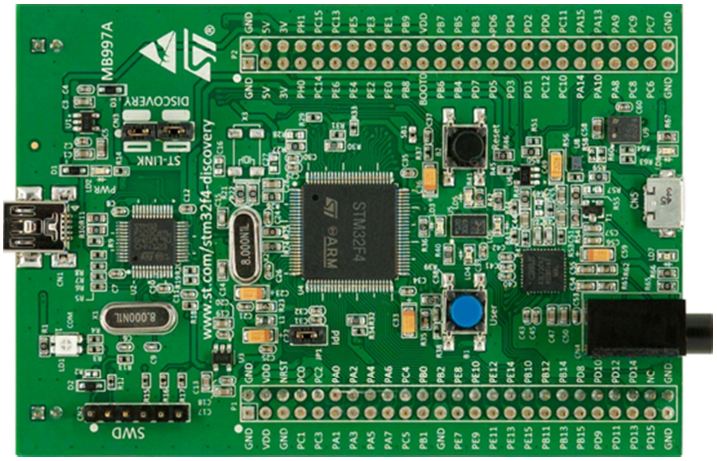


Figure 1: STM32F4Discovery board.

MCU	STM32F407VGT6
Family	ARM Cortex-M4
Vendor	ST Microelectronics
Package	LQFP100
RAM	192Kb (128Kb RAM + 64Kb CCMRAM)
Flash	1024Kb
Frequency	up to 168MHz
FPU	yes
Timers	14 (12x 16bit, 2x 32bit [TIM2 + TIM5])
ADCs	3x 16 channel 6 to 12-bit
UARTs	6
SPIs	3
I2Cs	3
Vcc	2.0V - 3.6V
Datasheet	Datasheet
Reference Manual	Reference Manual
Programming Manual	Programming Manual
Board Manual	Board Manual

Table 1: STM32 General features

Both output and input voltage limits for the microcontroller are between 0 and 3.3 V therefore some conditioning circuit will be needed in order to adjust the signals to this range.

2.2 Laboratory models

Two laboratory models will be used: pneumatic system and motor. These two models have been used for a long time in laboratory classes and both have different setups and expensive specific hardware that have been used to show how to control the models and their responses.

2.2.1 Motor

This model was built around a DC motor that is powered by 24 V and can reach 4000 r.p.m. The main components of this model are an amplifier, an encoder and a tachometer, but it also has two potentiometers which can change the gain of the stages and a lever as it is shown in Figure 2. These elements can change the working conditions of the model allowing different scenarios.

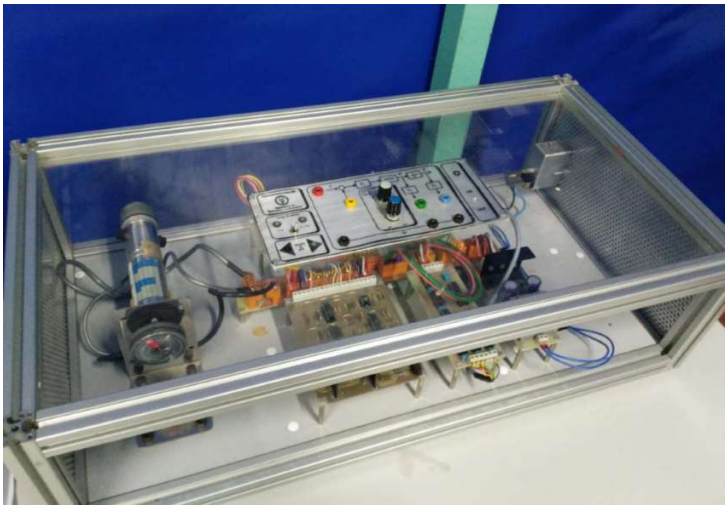


Figure 2: Motor Model

The amplifier

The main input of the model is the control signal but it needs to be converted in a power signal in order to move the motor. The amplifier converts the 0/10V or -5V/+5V control signal into a suitable signal to feed the motor. This stage also works as a comparator when there is a feedback in the control system.

The encoder

The main output of the motor is the spinning velocity of the motors shaft, with this velocity the encoder provides a signal proportional to the real angular position of the shaft which is the integral of the velocity and the models output. In the Figure 3 a feedback control system is implemented and it is shown how the position provided by the encoder will be compared to a reference signal, giving as a result the control input for the model.

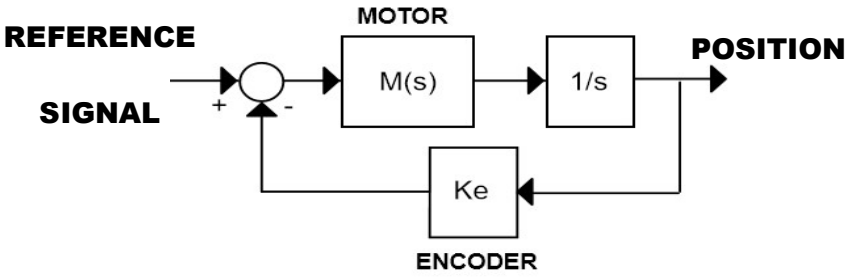


Figure 3: Feedback control example

The tachometer

The tachometer converts the angular velocity of the shaft into a signal that is proportional to the real velocity of the motor, generating another parameter in which control has to be performed.

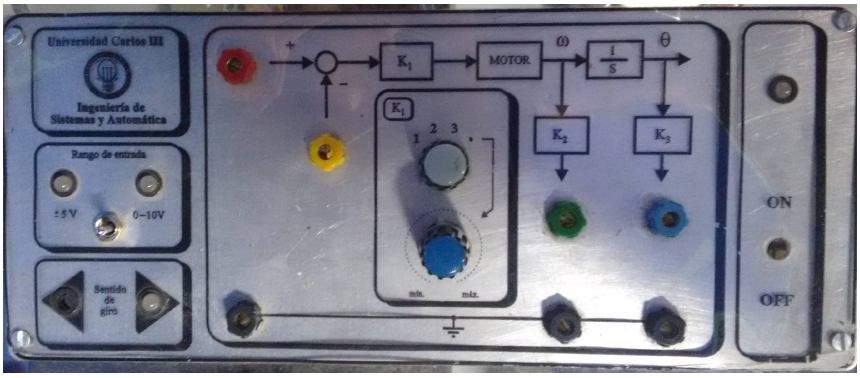


Figure 4 : Motor model connection board

As it is shown in Figure 4 the connection board of this model has two outputs and one input which can change the range with the lever. Signals between 0V and 10V or between -5V and +5V will be used to feed the input. The range of the two outputs depends on the two potentiometers that change K_1 gain, but the outputs are limited by an internal circuit that saturates when the voltage reaches 20V or -20V which establish the output voltage range.

2.2.2 Pneumatic system

This model performs as a pneumatic actuator with a load that has to be moved and controlled. As it is shown in Figure 5, the compressor feeds the cutoff valve that provides the air to the servovalve which controls the pneumatic actuator. The main components of this model are the servovalve and the linear potentiometer that is attached to the actuator.

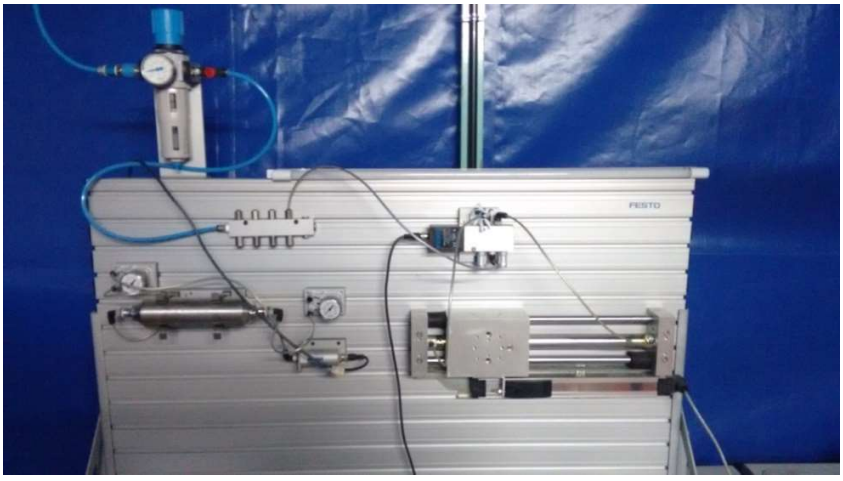


Figure 5 : Pneumatic system model

The servovalve

The servovalve, Figure 6, is the beginning of the control system, in this component the analog control signal is transformed into the opening or closing of the sections that moves the actuator. Servovalve's input signal must be between 0V and 10V in which both represents the extreme position of the valve and 5v stands for a full close in both sides of the valve.

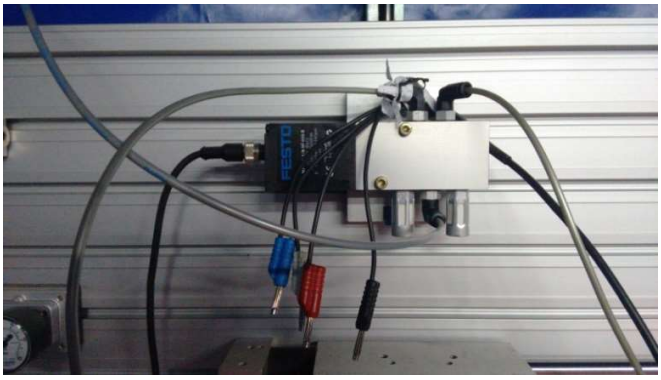


Figure 6: Servovalve

The linear potentiometer

This element acts as a sensor in the model. A sensor signal is created between the feeding voltage and the position of the actuator. This signal range will be between 10V and 0V.

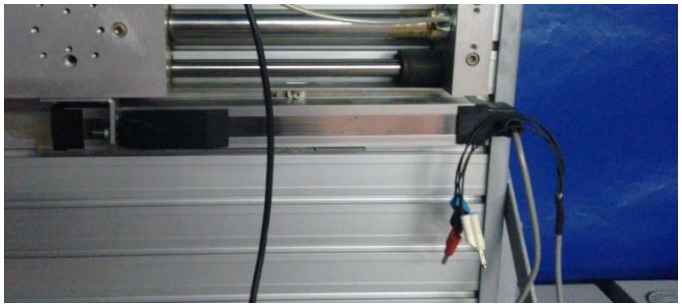


Figure 7: Linear Potentiometer

As it is shown in Figure 6 and Figure 7 these Festo systems have four color coded wires to perform the connections. In both cases Red is for the main power source which has to be 24V, Blue is for the sources GND, Black is for the control signal and White is for the signals GND.

The linear actuator

It consists of a double acting pneumatic cylinder rodless 200 mm stroke and 25mm diameter piston. The cylinder chambers are fed with compressed air by two connections. Potent permanent magnets attached the inner plunger with the slide. The travel speed is limited to keep it from exceed the strength of the magnetic coupling.

3. Laboratory practices board

This section covers the design and development of the prototype running through each step of its evolution, from the required components and schematic design using Altium [1] to the actual fabrication of the prototype itself.

3.1 Prototype design

The initial idea for the control system can be illustrated by the following diagram, Figure 8.

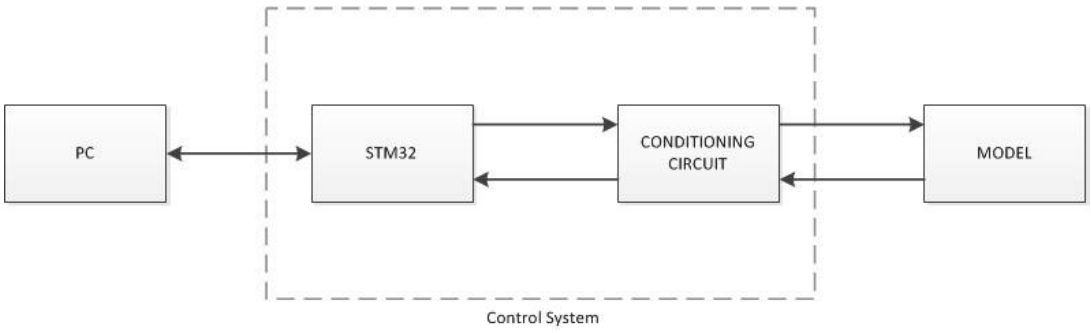


Figure 8: Control system diagram

The core of this system will be the microcontroller which communicates with the PC and generates the control signals. Since the voltage range in which the controller works is very limited, 0- 3.3V, the necessity of a conditioning circuit arises. The main function of this conditioning circuit will be to expand the voltage range for both input and output of the microcontroller, and in order to do so it is required to add and multiply the signals. Taking into account the models specifications described in Chapter 2, a first

prototype was designed. At this point the concept of a generic control system aroused and the design of a versatile, resistant, simple and economic conditioning circuit started.

3.1.1 Components

To fulfill the design a few components will be indispensable. To develop almost any operation with signals an operational amplifier or op-amp is used and there are all kinds of specific op-amp which are built to offer a better performance, but the "general purpose" ones offer more flexibility at a lower price. One of the most common will be used to design the conditioning circuit, the TL082 [5] which has 4 op-amps in a dip14 package. In addition to the op-amp, one or more voltage regulators such as LM317[7] will be needed to generate offset signals that allow the microcontroller to work with negative voltages.

3.1.2 Proposed Solution

Once seen the main components and the standards of flexibility and adaptability required, a solution that satisfies all the criteria is presented. Figure 9 shows a simplified diagram that explains the details of the solution:

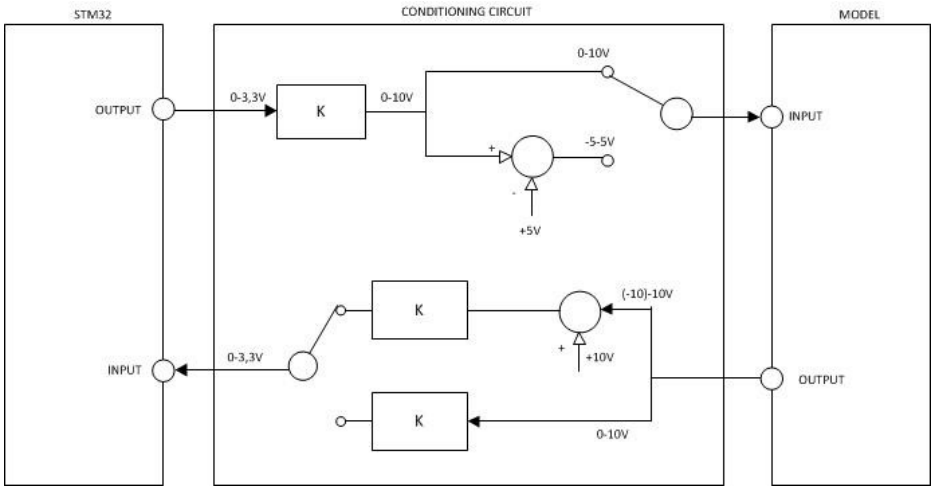


Figure 9: Proposed Solution diagram

The solution is a PCB (Printed Circuit Board) with the smaller dimensions possible that will contain the conditioning circuit. This circuit will be able to transform the output signal from the microcontroller into a 0/10V or -5v/+5V. Additionally, the circuit will allow the microcontroller to read input signals from 0/10V or -10V/+10V.Both output and input signals range will be chosen by auctioning a lever embedded on the circuit.

3.2 Prototype development

In this section the development process for the proposed solution will be explained in detail, from the schematic, through the board routing and fabrication to the soldering of the components and finishing touches. To develop the prototype an electronic design software has been used, Altium Designer. The following sections will feature some of the most outstanding characteristics of this software giving its importance in the designing process of this project.

3.2.1 Schematic

The first step is to create a new “PCB Project” in which you can create any number of schematics and PCB. The appearance of the software with a blank schematic is shown in Figure 10.

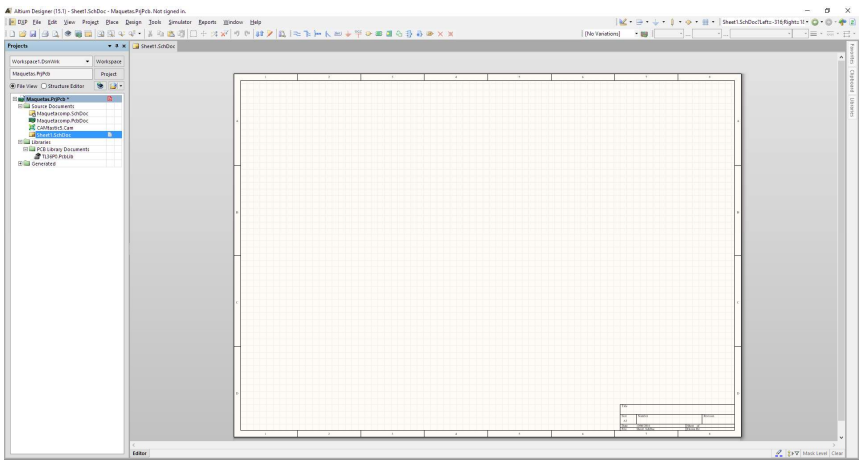


Figure 10: Blank Altiums schematic

Once a blank schematic is created, parts can be added to it as long as they exist in the libraries. There are standard libraries [2] which have ordinary components such as resistors, capacitors,

header, etc. Also almost every electronics brands have specifics libraries for their products. These libraries define everything: the symbol used in the schematic, the number and thickness of drills, how the parts behave in simulation, etc. However a specific part can be created by designing the footprint which can be done by hand inside the program, using “Footprint Editor” to make every aspect of the component or with the “Component Wizard” that asks for relevant parameters to autocreate the footprint. One of the most impressive features of Altium is the 3D view generator, every part or even full boards can be inspected in 3D. In Figure 11 and Figure 12 a footprint is being edited both in 2D and 3D. In this prototype every footprint used are surface mount technology or SMT which decreases the size of the parts and the board. This technology is also the easiest to solder and the most used.

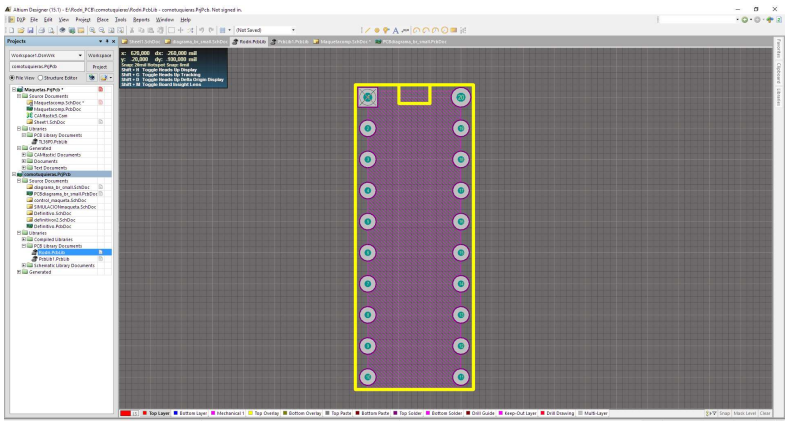


Figure 11: Altium Footprint Editor

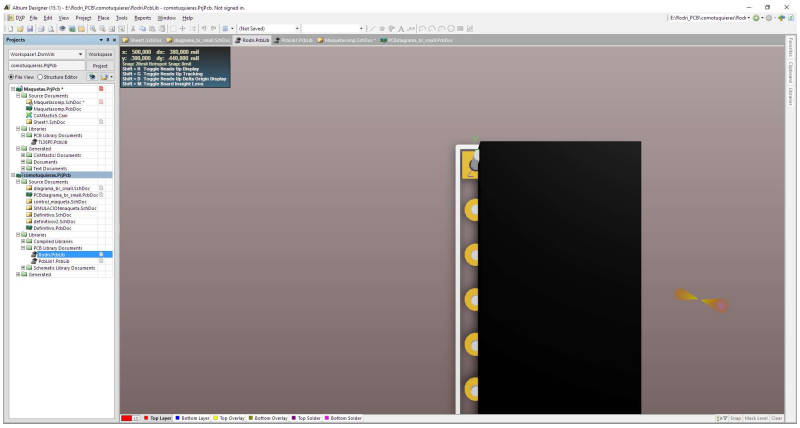


Figure 12: Altium Footprint Editor 3D mode

Now the full schematic is going to be explained starting with the voltage regulators. Since one of the desired ranged for the input is -10/+10 an offset of +10V is necessary to ensure that the microcontroller does not receive any negative values. Likewise there is an output that has to reach -5V. These offsets are being obtained through LM317 and LM337[8] voltage regulators, Figure 13. These specific components are exceptionally easy to use and require only two external resistors to set the output voltage. In this case the output will be regulated by a potentiometer that can be manually adjusted to compensate the real loses if needed.

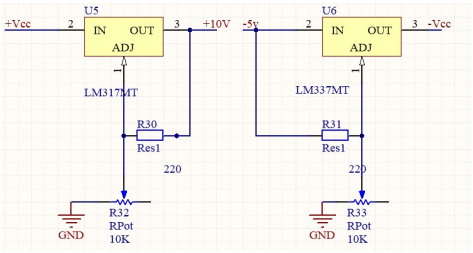


Figure 13: Voltage regulators schematics

The output conditioning circuit receives a signal from the microcontroller that can vary between 0 and 3.3 V. As it is shown in Figure 14, the circuit has two differentiated stages. In the first part a 3.3 gain is applied giving a 0-10V signal as a result while in the second one an offset of -5V is added. A mechanical switch with a lever is placed before the output to choose between the outcomes of the two stages, 0/ 10 V or -5/ +5 V. A single properly decoupled by a 100nF capacitor TL084 chip provides the 3 op-amps needed and all the resistors values were theoretically picked.

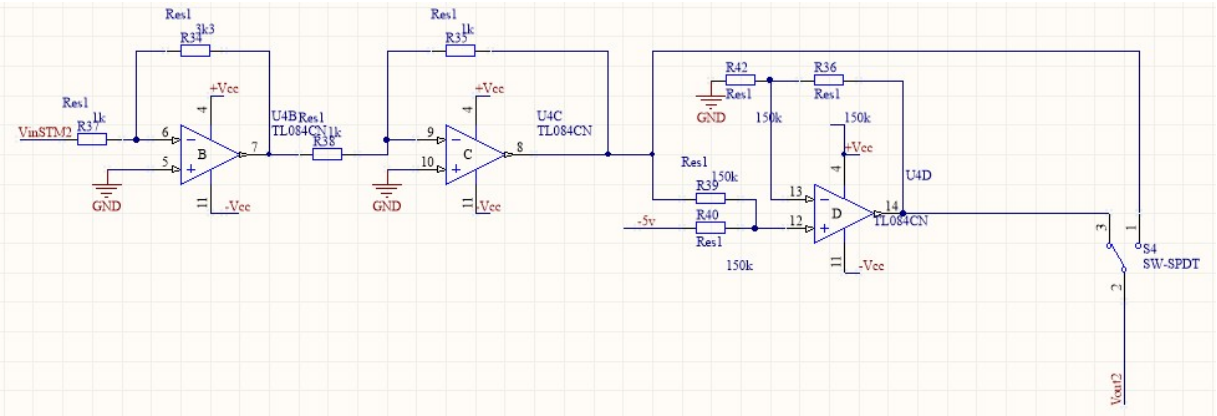


Figure 14: Output conditioning circuit

The Input conditioning circuit receives a signal from the model that can vary between -10 and 10 V or between 0 and 10V depending on the model. As it is shown in Figure 15, the circuit has two clearly differentiated ways one for each input range. The signal is fed to both paths simultaneously. The first one which gives as a result vM1,have two stages, first a 10V offset is added to the feed turning it into a 0/20 V signal. After that a 1/7 gain is applied to suit the microcontrollers input range. The second path only applied a 1/3 gain to the signal and a mechanical switch with a lever is placed before the output to choose between the outcomes of the two ways, -10/ +10 V or 0/ 10 V . Two properly decoupled by a 100nF capacitor TL084 chips provide the 5 op-amps needed and all the resistors values were theoretically picked.

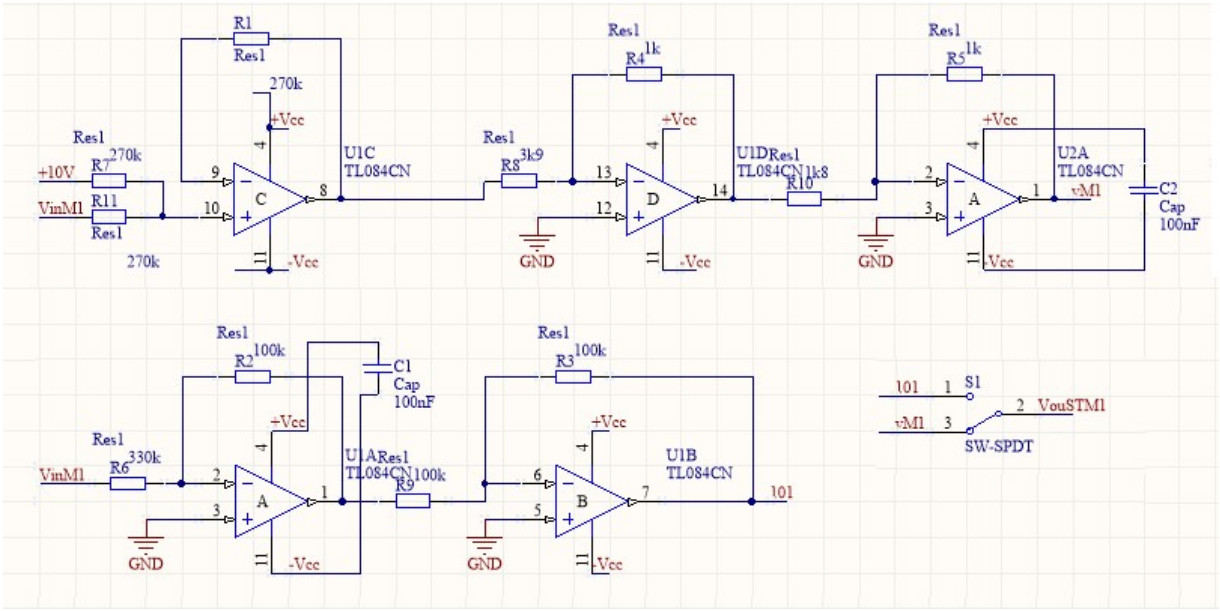


Figure 15: Input conditioning circuit

Finally, Figure 16 shows a full schematic in which the duplicity of both input and output circuit is clear. This duplicity allows controlling two different systems or two outputs from a single system. It also shows the headers that will be implemented on the board. One to power the circuit with +15V,-15V and GND, and the other two will be the links between the conditioning circuits, the microcontroller and the

models. These headers will be inputs, outputs and GND which will be common for all the components and the models.

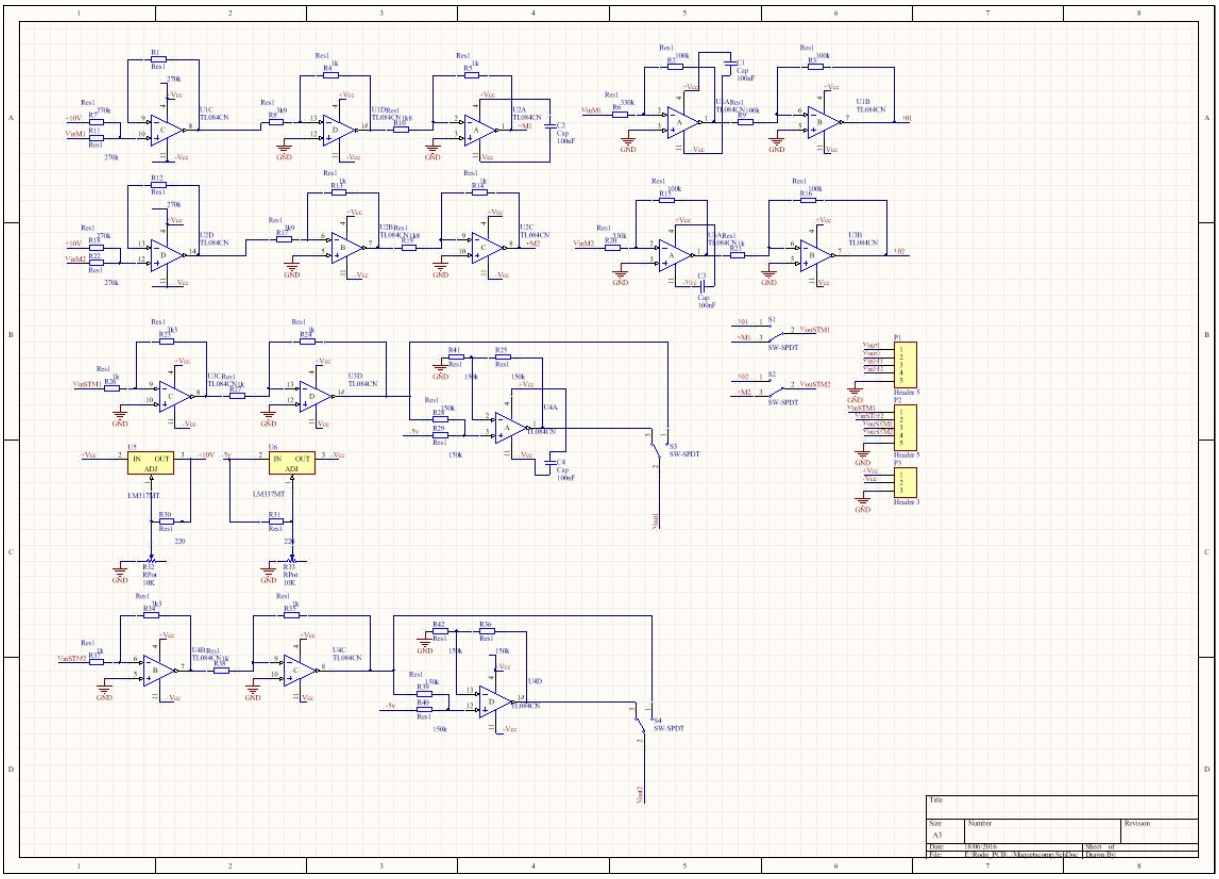


Figure 16: Full Schematic

The last step is to compile and search for errors in connections or misplaced net labels. Also it is really important to annotate correctly, the program can do it automatically but in cases like this when you have shared chips is easier to route if they are close to each other.

3.2.2 PCB Routing

The first step is create a “PCB” file in which by using the option “Update PCB Document” all our footprints and their connections are going to appear. As it is shown in Figure 17 all the parts have to be manually placed trying to make the tracks as short as possible.

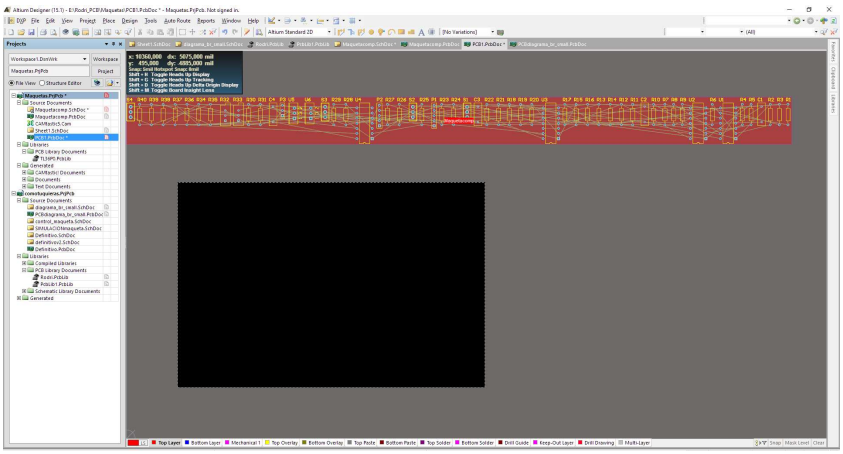


Figure 17: Altium PCB workspace.

Since this PCB is going to be built based on SMT, all the components will be placed on the Top layer. Layers Bottom and Mechanical 1 will be used to route and to delimitate the physical space of the board. Once the components are more or less well distributed on the board, Figure 19, the routing rules must be defined.

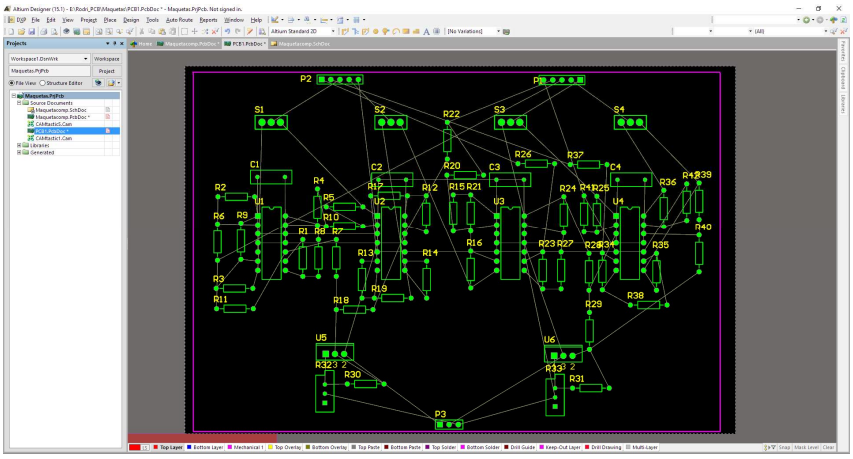


Figure 18: Un-routed placed components.

To set the rules in Altium go to Design/ Rules… a window like the one in Figure 19, with all the routing options will appear. This is the most important part of the routing because is when all the parameters are going to be defined: track width, distance between components, clearance, drills, etc. The software will warn you if one of these rules is being broken.

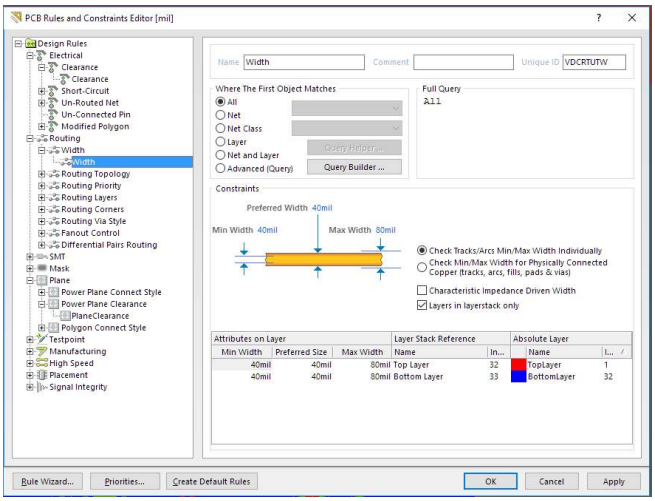


Figure 19: PCB Rules Editor.

In this case the PCB has been design with 40mils of track width and 30 mils of clearance. As it is shown in Figure 20, it has a GND plane on Bottom layer and has been routed following the Bus topology for the power sources tracks.

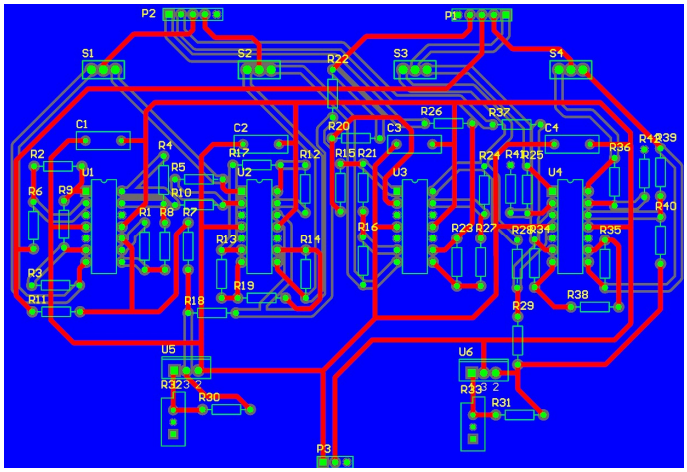


Figure 20: Routed PCB

Once finished it is verified that all the pre-established rules of design are fulfilled by means of the option “Design Rule Check” of the menu “Tools”. This step is crucial because taking into account the routing rules previously set a list of errors is deployed which allows an easier debug of the PCB.

3.2.3 Manufacturing PCB

As soon as the routing process was finished the files necessary for the manufacture of the PCB had to be generated: the Gerber files “.gbr” of the top, low and outline layers which are used to draw the tracks by the milling machine, and a text format file “.txt” that contains all the necessary information about the drills in the PCB. To create these files Altium has the menu “Fabrication Output… ” in which an output file can be selected from a long list. In this case, Gerber Files and NC Drill Files are chosen.

These files are used by the milling machine to construct the PCB. As it is shown in Figure 21 and Figure 22, which correspond to the Top layer and the Bottom layer, the result is a 100 mm x140 mm board design to be coupled with the STM32F4Discovery.

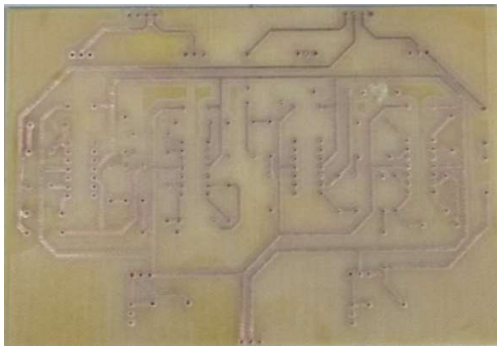


Figure 21: PCB Top layer

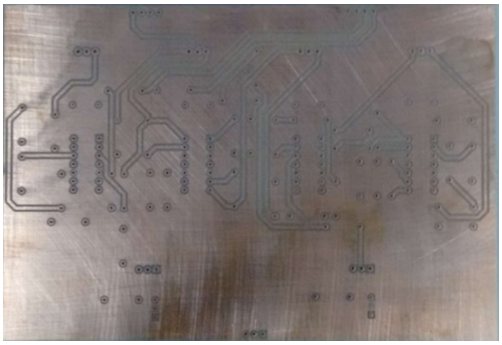


Figure 22: PCB Bottom layer.

With the already built PCB and after verifying that it has no manufacture errors nor any open track or cut circuit, the phase of assembly began. Using Altium's "bill of materials" Table 2 which includes all the components, their values, quantities and designation has been created.

Description	Designation	Value	Quantity
TL084	U1,U2,U3,U4		4
LM317	U5		1
LM337	U6		1
TL36P0	S1,S2,S3		3
Header 5	P1,P2		2
Header 3	P3		1
R Potentiometer	R32,R33	10K	2
R power	R30,R31	220	2
Resistor	R1,R7,R11,R12,R18,R	270k	6
	R25,R28,R29,R36,R3	150k	6
	R8,R17	3k9	2
	R6,R20,R23,R34	3k3	4
	R10,R19	1k8	2
	R2,R3,R4,R5,R9,R13,	1k	10
	R24,R26,R27,R35,R3	1k	6

Table 2: Bill of materials

Before assembly, the PCB is impregnated with a substance that prevents from oxidizing and facilitates the welding since it contains flush. To solder all the components a welding station that has all the necessary hardware to carry out this project has been used. The order followed to place the components gives priority to those that takes a major access difficulty such as the smallest ones or the ones who are closed to each other's. Figure 23 shows the board after the installation of all the components.

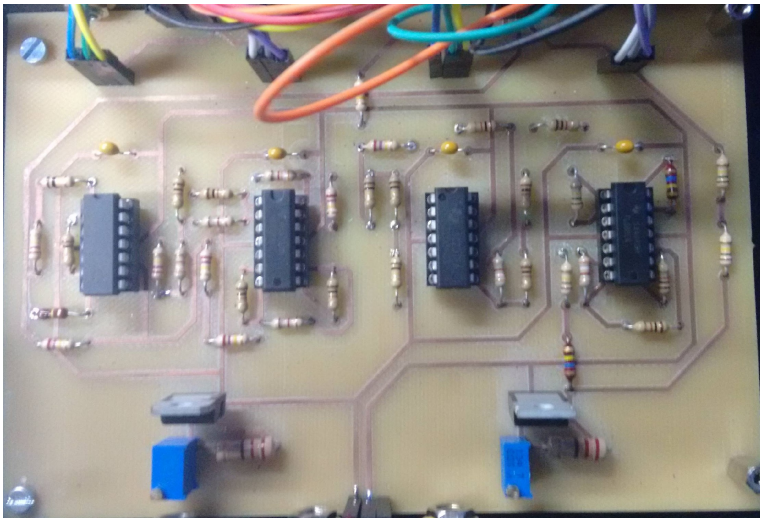


Figure 23: Fully assembly PCB

In order to protect and close the system a 210 mm x 110mm x45 mm black PVC box was purchased. Since the box was design for general purpose some adjustments were necessary. Five drills were placed in one of the large sides to fit the connectors for the control signals. Additionally three more drills for the main power source connectors were drilled on the opposite side. Two USB connector size holes were carved to grant access to the STM32F4Discovery and finally the cover was drilled to place the four switches. Figure 24 shows the box open with all the connections made.

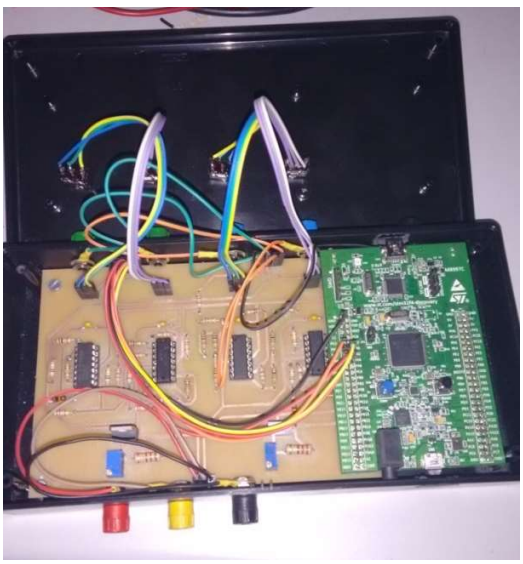


Figure 24: Finished system with the box open.

Finally both boards were glued to the box to prevent unnecessary movements or even disconnections and all the external connectors were labeled as it is shown in figure 25.

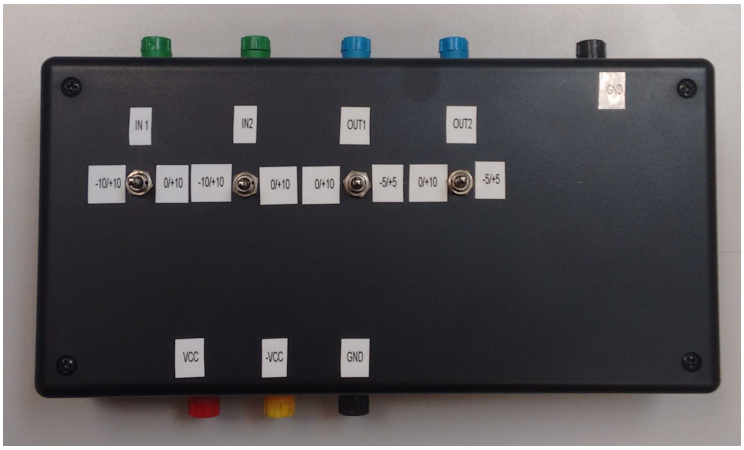


Figure 25: Finished system with the box close.

4. Electronic Speed Control

In this section an Electronic Speed Control is going to be described, classified and programed. The ESC is an electronic circuit with an input channel and three outputs as it is shown in the Figure 26, built to control an electric motor's speed, its spinning direction and in some cases even acts as an electric brake. Mostly used in RC modeling, ESC systems vary the switching rate of a network of field effect transistors (or FETs) to both control the motor and acquire control information.

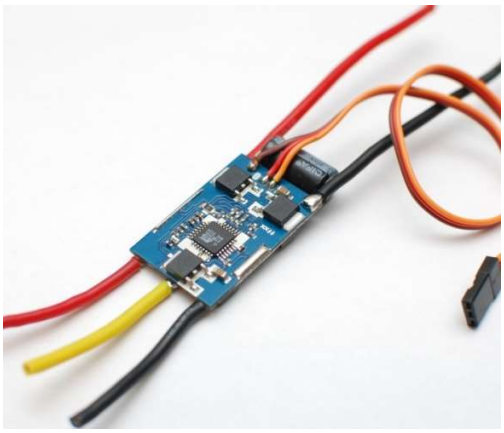


Figure 26: ESC

4.1 ESC types.

There are two kinds when we look at the way they connect to the receiver: a "stand-alone" unit which plugs into the receivers throttle control channel or incorporated into the receiver itself which

allows the receiver to control more than one motor and also minimizes the size (really important for small drones). A “stand-alone” model will be used in this project.

ESC systems are mostly used on brushless electric motors but there are ESC for brushed electric motors although they are very different by design and therefore not compatible.

4.1.1 Brushed motor

The ESC supplies the power with short pulses, many times a second. The speed is controlled by changing the width of the pulse since the motor receives more power when the pulse is wider. Brushed motors connect to the speed controller via two wires and the only way to reverse the direction of the motor is swapping the wires.

4.1.2 Brushless motor

The ESC for a brushless motor is completely different and more complex. Figure 27 shows how controlled by the signal feed to the microprocessor it creates a tri-phase AC power output of limited voltage from a DC power input using field effect transistors. The ESC has to take into account that the correct phase varies with the motor rotation. To detect this variations some ESC that use magnetic(Hall Effect) or optical detectors, but most modern brushless speed controllers are known as “sensorless controllers” and make use of the voltage generated in the momentarily unpowered windings to determine the position of the motor(back EMF).In this project we will use a sensorless electronic speed controller for brushless motor.

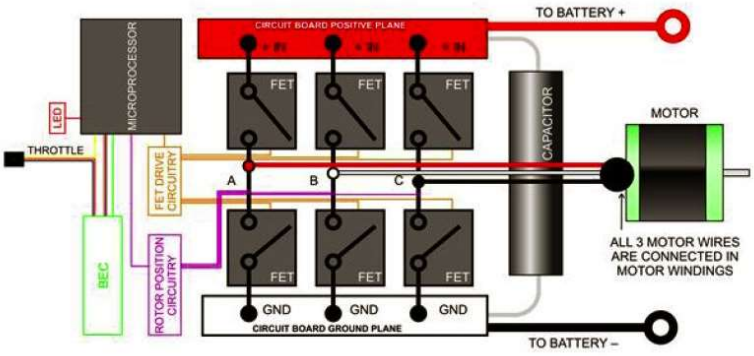


Figure 27: Brushless ESC working diagram.

4.2 ESC firmware

Most modern ESC contains a microcontroller that reads the input signal and generates the appropriate control signals using a built-in program or firmware. In some cases it is possible to change this firmware, upgrade it or simply replace it. In some cases this firmware is programmable by using a computer, a specific programming-card or even a broadcast station. Programmable speed controls generally have user-specified options which allow setting low voltage cut-off limits, timing, acceleration, braking and direction of rotation.

4.3 PULSO DLC 30A

Once realized how many different kind of ESC exists and what features were essential to us in order to achieve the goals of this project: Forward/Brake/Reverse mode, “Stand-alone”, sensorless and programmable firmware. This specific ESC caught our attention, because it fulfills all the specifications and there was great reviews of it performance for radio control racing cars. It has no datasheet since it is a close system but the user´s manual have all the information we needed and also the instructions on how to program the firmware.

4.3.1 Features

- It (DLC 30A) is designed for only Model Car with the operating voltage 4V-14V.

- Automatic power cut-off. (When the radio signal loses for more than 3 seconds, the power will automatically be cut off.)

- BEC (5V/2A) provides power to receiver and servos.

- Over-heat protection. The power will be cut-off as it is heated up to 110º.

- Programming-Box (Prog-Box) can make ESC DLC 30A setting easier and extend the possibilities of ESC settings.

4.3.2 Factory Default Setting

- Forward/brake/reverse mode.

- Timing mode 1 (0-7°).

- Cut-off voltage mode 1 (Intellectual cut-off for Ni-MH/Ni-CD battery).

4.3.3 Operation—For DLC 30A without Prog-Box

At this point is important to make a clear differentiation between programming the firmware and calibrating the ESC, because programming an ESC without Prog-box or USB connection requires a broadcast station and a receiver which has to be calibrated in order to work properly. Pulso DLC ESC uses two PWM control signals: one generated when the radio throttle stick is at maximum position, “full throttle”, and one generated when the radio throttle stick is at mid position, “close”. The first step is connect the brushless motor to the ESC because it uses the motor to generate audible acoustic signals or “beeps” to guide us through programming. In order to learn how the ESC works, a broadcast station and a receiver was used to map the control signals. As it is shown in Figure 28 “full throttle” signal has a significant wider pulse and a slightly higher Vmax which generates a difference of 100mv on the Vmean. It is also noticeable that both signals have the same period.

Figure 28 : “Full throttle” ESC signal on the left. “Close” ESC signal on the right.

Calibrate the Pulso DLC.

Figure 29 graphically describes the process to calibrate the Pulso DLC. The first step is to enter the calibration mode, by doing this the ESC starts to read and save the main signals “full throttle” and “close”. Once read these specific signals will control the ESC and won’t change until another calibration.

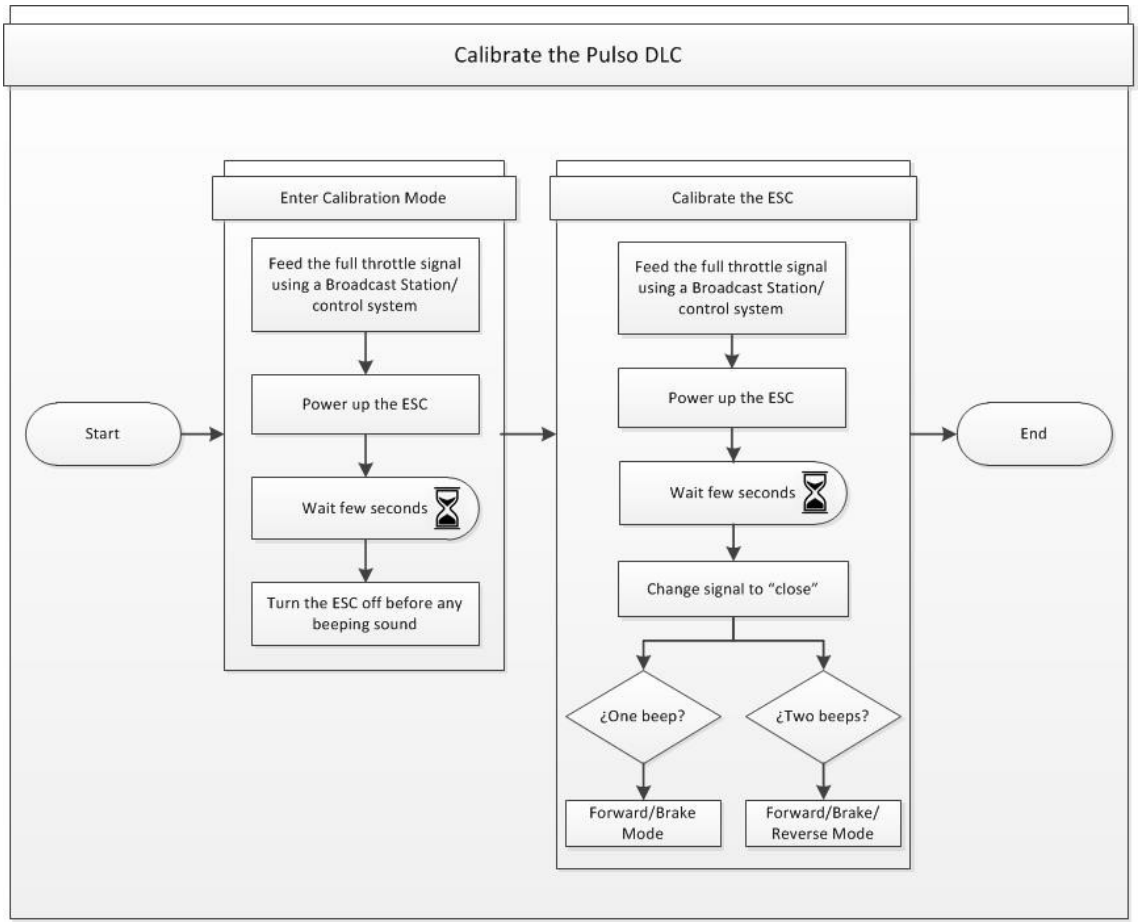


Figure 29: Pulso DLC calibration.

Program

Program the Pulso DLC is a linear tedious process, the program mode is always the same: “full throttle” signal has to be feed before turning on the ESC and during the whole program process. In order to inform which feature is modifying it emits a series of beeps that change depending on the mode. To select and successfully program one feature the fed signal has to change from “full throttle” to “close”. Each feature has to be programmed one by one since the process ends with the change on the input signal. Figure 30 graphically describes the programing process.

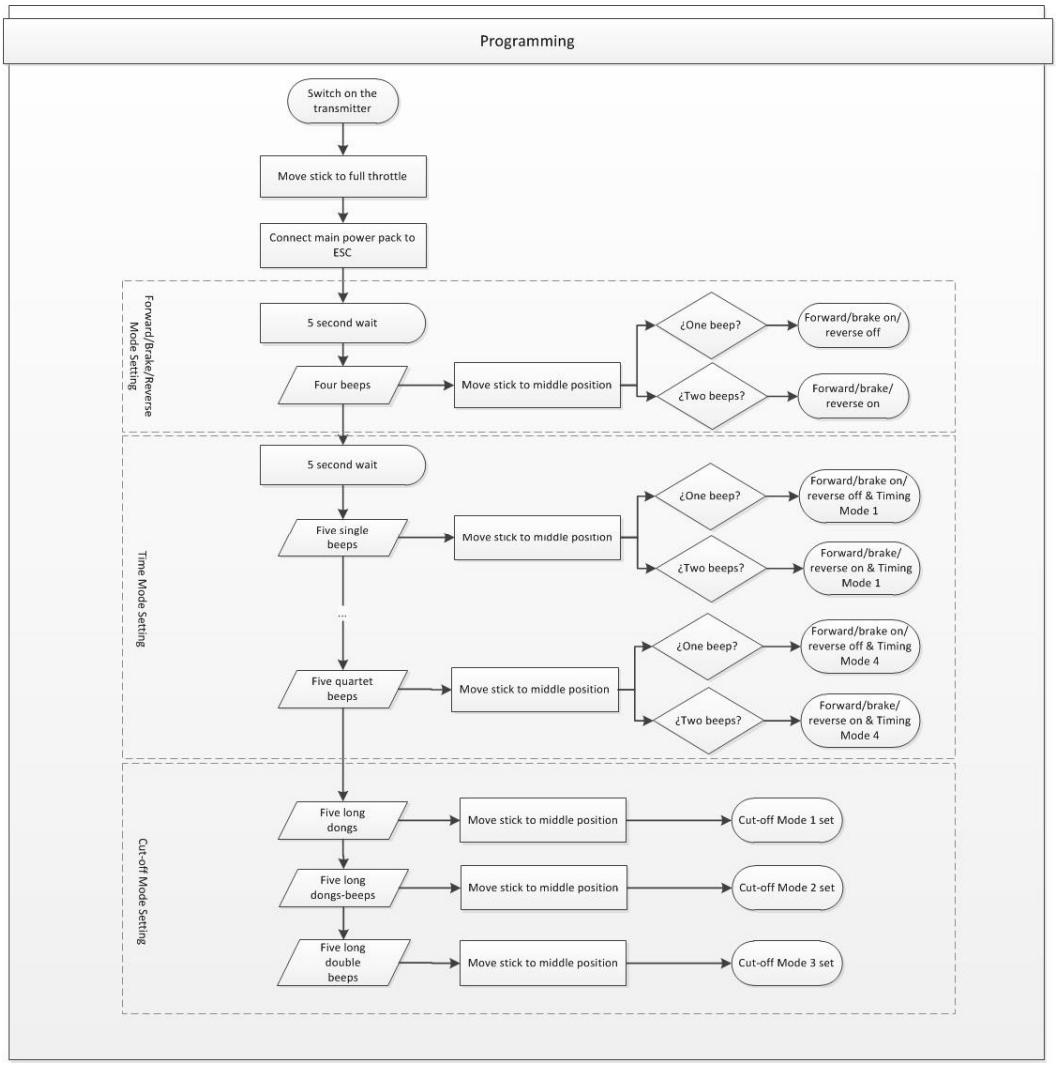


Figure 30: Pulso DLC Program mode

Forward/brake/reverse or Forward/brake mode:

This feature changes the mode in which the motor works, it is really important because depending on the RC model brake or reverse mode can even crash a drone or a helicopter.

Timing mode:

The timing mode allows the ESC to adjust better to a specific brushless motor. It's the only parameter that takes into account the motor and how it is build. Since every motor is different Pulso DLC offers 4 different timing modes from 0º to 30º. Table 3 below contains a rough approximation of the relation between the number of poles of a motor, the timing advance and the performance.

Mode	**Low Timing Advance (Timing Degrees – 5 & 10)**		**Standard Timing Advance (Timing Degrees – 15 & 20)**		**High Timing Advance (Timing Degrees – 25)**	
Motor Poles	2 to 4	6 or more	6 to 12	14 or more	12	14 or more
Expected Performance	Good balance of power and efficiency	Best efficiency and run time (lowest power)	Good balance of power and efficiency	Best efficiency and run time (lowest power)	Highest power, less efficiency	Good balance of power and efficiency

Table 3: Relation between Timing Advance and motor poles.

Voltage cut-off mode:

The cut-off voltage protects the battery from an excessive discharge, every battery has a recommended cut-off voltage and it can vary very much between models. In this project we will not use any battery but is an important feature that can be programmed.

5. Experimental results

In this chapter all the tests done to the hardware will be exposed along with the results obtained. To properly test the laboratory models and ESC some Matlab [11] diagrams and Arduino programs have been developed and will be explain too in the next pages.

5.1 PCB Test

After manufacturing the PCB to test how it worked was essential. The first step was to power up the circuit and check all the critical points of the circuit with the help of a multimeter, such as power feeds for the TL084 chips, known voltages drops and all the connections to GND. The next step was to feed some control signals and check every op-amp operation, calibrate the potentiometers and check the output signals. To perform the tests a signal generator and a Tektronix TDS 214B oscilloscope have been used.

The first way tested was the input to the model. According to the design from Chapter 3 it should convert a 0-3.3V control signal emitted by the microcontroller into a -5 V/+5V or 0/10V. To start the test a 0-3V square signal was fed through the track. Using two oscilloscope probes, Ch1 as input and Ch2 as output Figure 31 was obtained. On the left -5 V/+5V conditioning circuits output and 0/10V on the right.

Figure 31: Signal emitted by the Micro and conditioning circuit outputs,

Both outputs fulfil the design from Chapter 3 although there are some variations when compared with the theoretical values. These discrepancies are due to the tolerance in resistors and the small loses in the real op-amps.

To test the input from the model a different set of signals were implemented. According to the design from chapter 3 it should converts a -10/+10V or 0/10V signal from the model into a 0/3.3 V signal that can be read by the microcontroller. Additionally since the model can send any signal between those values the impact of this signal on the circuit have to be checked.

To start the test a 0-10V square signal was feed through the track. Using two oscilloscope probes, Ch1 as input and Ch2 as output Figure 32 was obtained; as it is shown the gain of this channel is a little bit high giving us a signal very close to the saturation area. This can be easily fixed by replacing some resistors.

Figure 32: Response of the conditioning circuit to a 0/10 V input

After that the -10V/10 V was tested. To start the test a -5/5V square signal was fed through the track. Using two oscilloscope probes, Ch1 as input and Ch2 as output Figure 33 was obtained; as it is shown the transformed signal is between the desired ranges of the microcontroller and the Vmean is 829 mV, this value should be fixed for any input and it shows that our gain is a little low which reduces the sensitivity of the system.

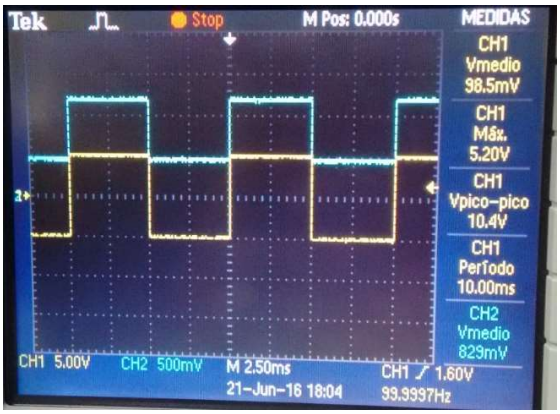


Figure 33: Response of the conditioning circuit to a -5/5 V input

Finally a -10/10V square signal was feed through the track. Using two oscilloscope probes, Ch1 as input and Ch2 as output Figure 34 was obtained; as it is shown the transformed signal is between the desired ranges of the microcontroller and the Vmean is 827 mV which probes the consistency of our system.

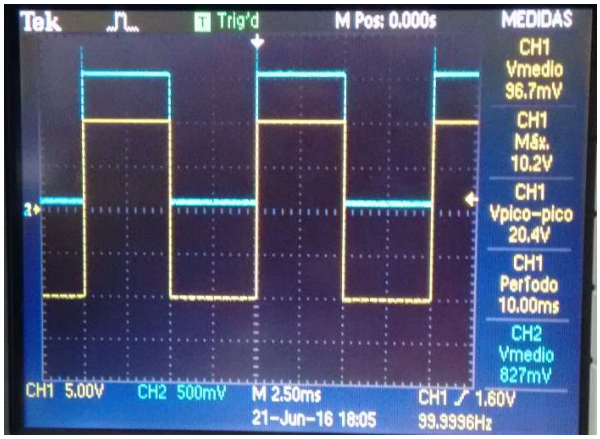


Figure 34: Response of the conditioning circuit to a -10/10 V input

5.2 ESC Test

As it was mention earlier the ESC have been tested using Arduino which is an open-source hardware and software specialized in microcontroller-based kits. The goal was to use a microcontroller to create the control signals for the ESC and taking into account that the control signals described in Chapter 4 are PWM signals Arduino was the perfect choice.

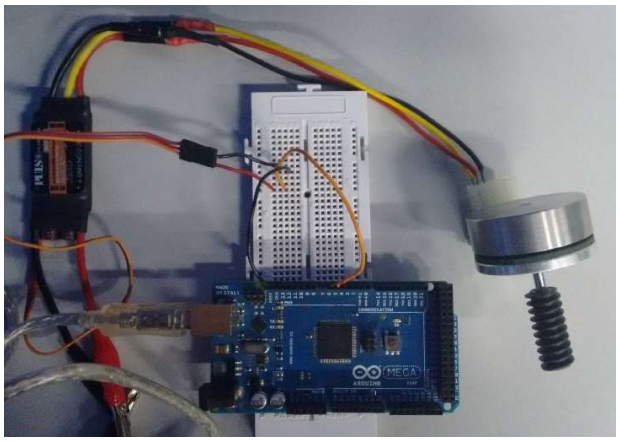


Figure 35: ESC controlled by Arduino assembly.

To begin the test a simple circuit was implemented to connect the ESC and Arduino, Figure 35. The three windings of the Maxton motor have to be connected to the ESC and a handmade connector was manufactured additionally the signal from the microcontroller have to be feed to the ESC input and

properly connected to GND. The goal of this test is to generate control signals that can control the ESC to move a brushless motor. First the code used to generate the control PWM is going to be described. Figure 36 declares the 3 variables that are going to be used: “Arming_Time” will be used to track time in order to calibrate the ESC, “pin” will be used to send the signal and “Pulse” will set the frequency.

```
int Arming_Time=0;
int pin =6;
int Pulse=1150;
```

Figure 36: Arduino variables

The first step is calibrating the ESC so it recognizes the signals from the microcontroller. As it was described in Chapter 4, in order to achieve calibration a “full throttle” signal is needed until the first 4 beeps. After that the ESC will read backwards and then calibration process will end with the “close” signal. This method depends highly on time which is why it had to be measured and then controlled in our code.

```
void setup() {
  pinMode(pin, OUTPUT);
  //will create a 1100us pulse to arm the ESC
  //the pulse will last for 20ms x 850 counts = 17s
   /for(Arming_Time =0; Arming_Time < 850; Arming_Time +=1){
      //FULL THROTLLE SIGNAL
      digitalWrite(pin,HIGH);
      delayMicroseconds(2000);
      digitalWrite(pin, LOW);
      delay(17-(Pulse/1000));
    }
      for(Arming_Time =500; Arming_Time <650; Arming_Time +=1){
      //BACKWARDS SIGNAL
      digitalWrite(pin,HIGH);
      delayMicroseconds(1000);
      digitalWrite(pin, LOW);
      delay(18-((Pulse)/1000));
      }
       for(Arming_Time =0; Arming_Time < 200; Arming_Time +=1){
      //CLOSE SIGNAL
      digitalWrite(pin,HIGH);
      delayMicroseconds(1500);
      digitalWrite(pin, LOW);
      delay(18-((Pulse+500)/1000));

      }
  }
```

Figure 37: Arduinos calibration code

Figure 37 shows a rudimental piece of code that was used to perform the calibration. Using the variable “Arming_Time” to take into account the time jumping from one loop to the next. Each loop

sends a different control signal starting with “full throttle” and ending with “close”. The signals are programed like a single step that repeats itself over time. The first delay establish the wide of the high pulse which is the one that defines the signal and the second delay adjust the frequency of the pulse. Finally a loop was implemented to keep the motor moving while changing the control signal, Figure 38.

```
void loop(){
     //FORWARD
      for (Pulse = 1150; Pulse <1350; Pulse+=1){
        digitalWrite(pin,HIGH);
        delayMicroseconds(2000);
        digitalWrite(pin, LOW);
        delay(17-(Pulse/1000));
      }
      //BRAKE
       for (Pulse = 1350; Pulse <=1400; Pulse+=1){
        digitalWrite(pin,HIGH);
        delayMicroseconds(1500);
        digitalWrite(pin, LOW);
        delay(18-((Pulse+500)/1000));}
      //REVERSE
       for (Pulse = 1400; Pulse >=1150; Pulse-=1){
        digitalWrite(pin,HIGH);
        delayMicroseconds(1000);
        digitalWrite(pin, LOW);
        delay(18-(Pulse/1000));
    }
}
```

Figure 38: Final loop

All the control signals were measured in the laboratory and although there are a little higher compared to the ones described in Chapter 4 the ESC works with them. Figure 39 represents the ESC control signals generated by Arduino, from left to right: “full throttle”, “close” and “backwards”.

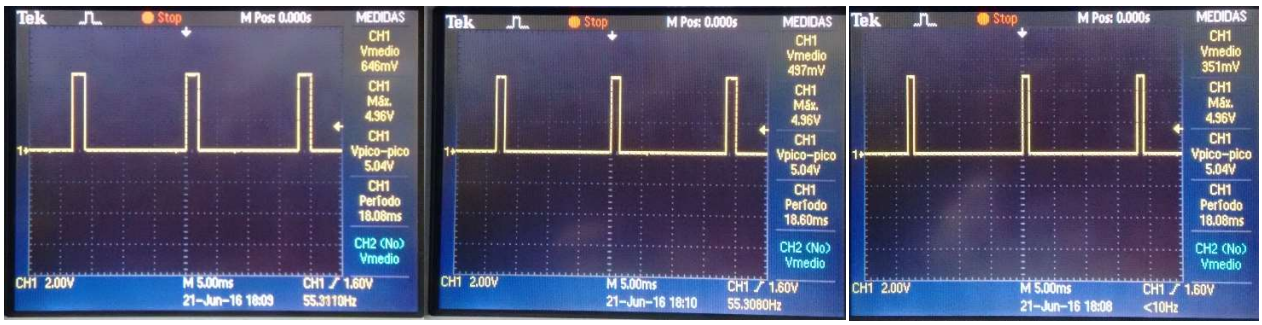


Figure 39: ESC control signals

5.3 Laboratory models tests

In order to test de control system a simple control software was developed using Simulink [12] block programming from Matlab Software. These block diagrams will be explain along with the experimental results obtained from each diagram in the following pages.

5.3.1 Matlab Simulink

Simulink uses the USB port of the computer to program in real time any microcontroller supported. To be able to do this a constant connection between the PC and the microcontroller is needed. The first step is to embed the Target block diagram into the microcontroller. This diagram configures the communications between the USB and the microcontroller pins. As it is shown in Figure 40 the diagram has 5 major blocks and 6 blocks that performs some escalating and data conversion. USB VCP blocks along with Target Setup belong to a library created by UC3M. These blocks are in charge of both emit and receive information through the USB.

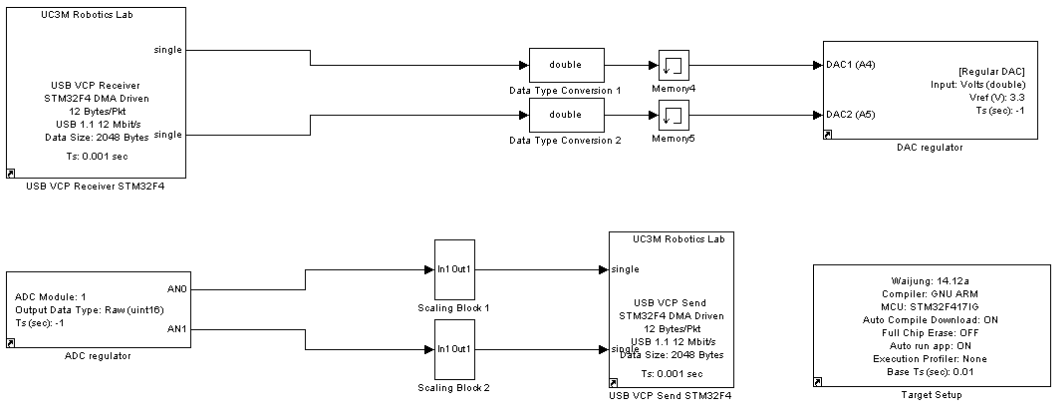


Figure 40: Target block diagram.

This block diagram will be used to perform every test since it will be embedded in the STM32F4Discovery microcontroller. To perform any test a block diagram in which set up the signals and check the data received from the USB port will be needed.

5.3.2 Motor model test

The first test for the model motor will be performed without feedback, the goal is to be able to read the signals from the tachometer and the encoder. A Simulink blocks diagram was implemented in order to read and send the control signal. As it is shown in Figure 41 Simulink generates a signal of -5/5V or 0/10V that goes into an escalating bock and it is sent through the microcontroller. At the same time a signal is being read and sent through the USB, after a minor scale the signal is compared to the input on a scope. After some testing a Real-Time block was added in order to diminish the time delay produced by the PC processing time.

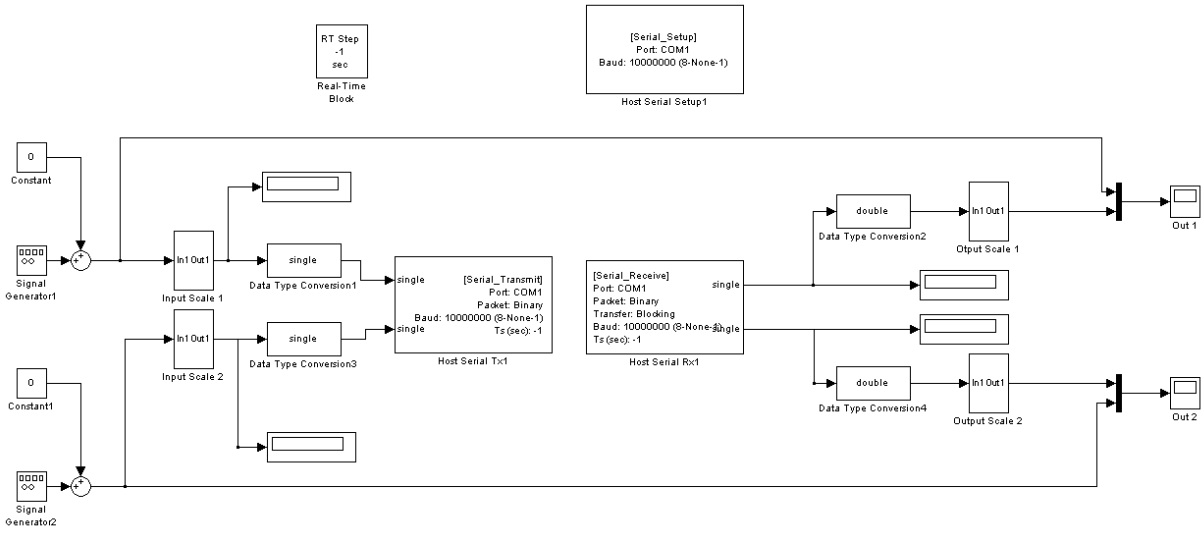


Figure 41: Simulink block diagram for motor test 1.

Host serials blocks are in charge of controlling the USB responses. These blocks belong to a library created by UC3M.

Figure 42 shows the motor response to a -5/5 V signal in both velocity and position for an open loop system.

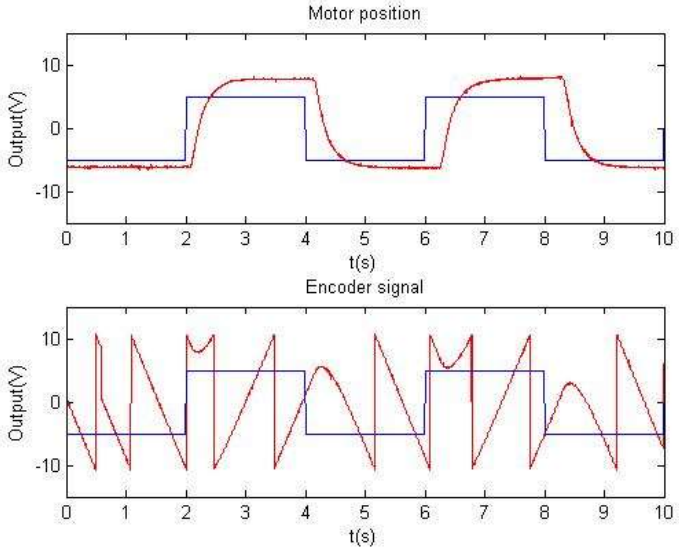


Figure 42: Motor model test 1 response.

Additionally a simple test were performed with the same Simulink scheme in which negative feedback in position was physically implemented on the motor model. As it is shown in Figure 43 in which magenta corresponds to the position of the motor and blue to the speed, a control state was achieve.

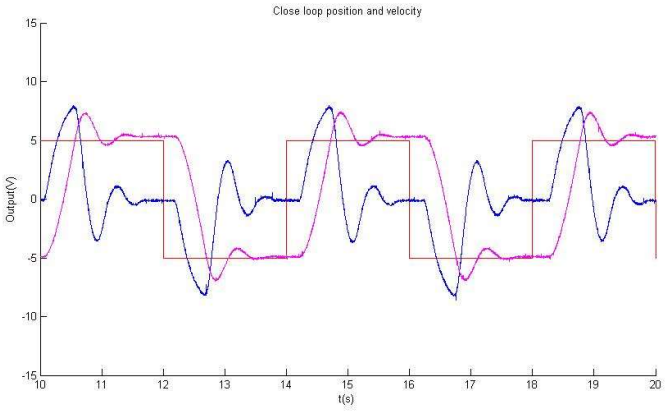


Figure 43: Motor model test 1 (feedback)

The second test for the model motor will be performed with negative feedback implemented within the Simulink diagram, the goal is to be able to control the motor model in position. A Simulink blocks' diagram was implemented in order to read and send the control signal. As it is shown in Figure 44 Simulink generates a signal of -5/5V or 0/10V that goes into an adder block in which the position signal is subtracted. After that a PID generates the control signal that with a minor escalate its ready to be sent through the microcontroller. After some testing a Real-Time block was added in order to diminish the time delay produced by the PC processing time.

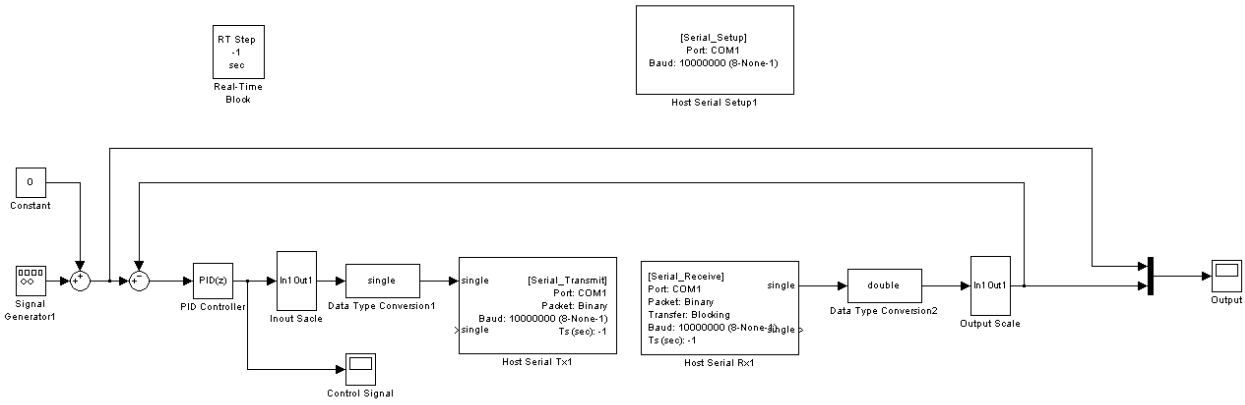


Figure 44: Simulink block diagram for motor test 2.

In order to successfully achieve control over the model the feedback has to be regulated by a PID. The goal of this test is to achieve some state of control but PID values have not been theoretically calculated since there was no intention to reach any specific control condition. Instead the values used have been practically found by trial and error. The final PID values used for the test were P= 1.2, I =2 and D=0.01. As it is shown in Figure 45 the response obtained was really close to a fully controlled system.

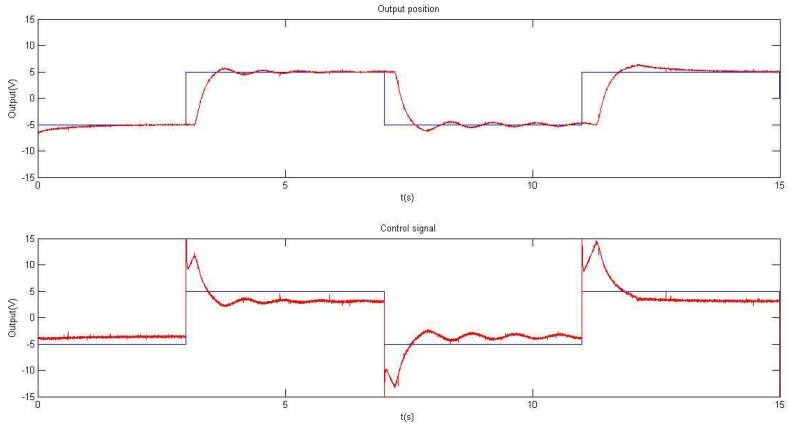


Figure 45: Motor model test 2 response

5.3.1 Pneumatic model test

The test for the pneumatic system will be performed with negative feedback implemented within the Simulink diagram, the goal is to be able to control the model in position. As it is shown in Figure 46 Simulink generates a signal of 3/8V that goes into an adder block in which the position signal is subtracted. After that a PID generates the control signal that with a minor escalate its ready to be send through the microcontroller. After some testing a Real-Time block was added in order to diminish the time delay produced by the PC processing time.

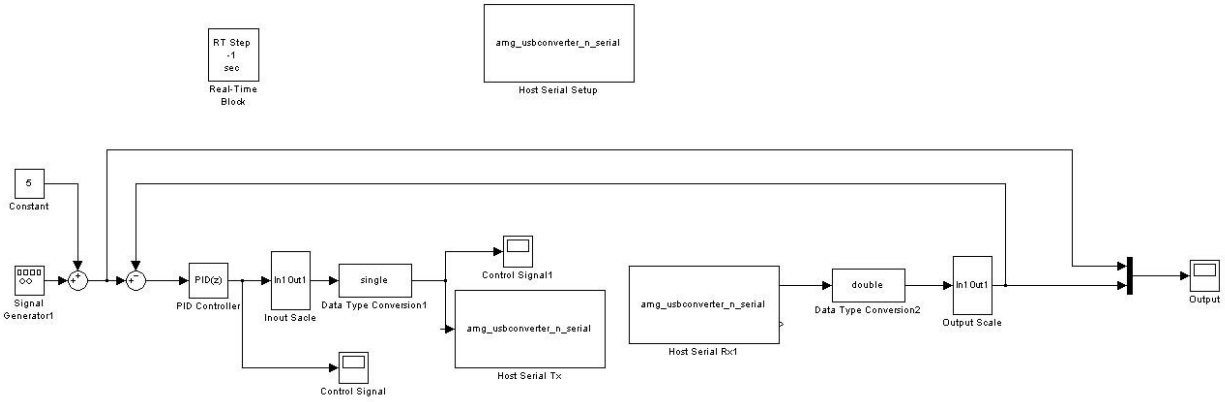


Figure 46: Simulink blocks diagram for pneumatic test

In order to successfully achieve control over the model the feedback has to be regulated by a PID. The goal of this test is to achieve some state of control but PID values have not been theoretically calculated since there was no intention to reach any specific control condition. Instead the values used have been practically found by trial and error. The final PID values used for the test were P= 0.3, I =0.9 and D=0.01. As it is shown in figure 47 the response obtained was really close to a fully controlled system.

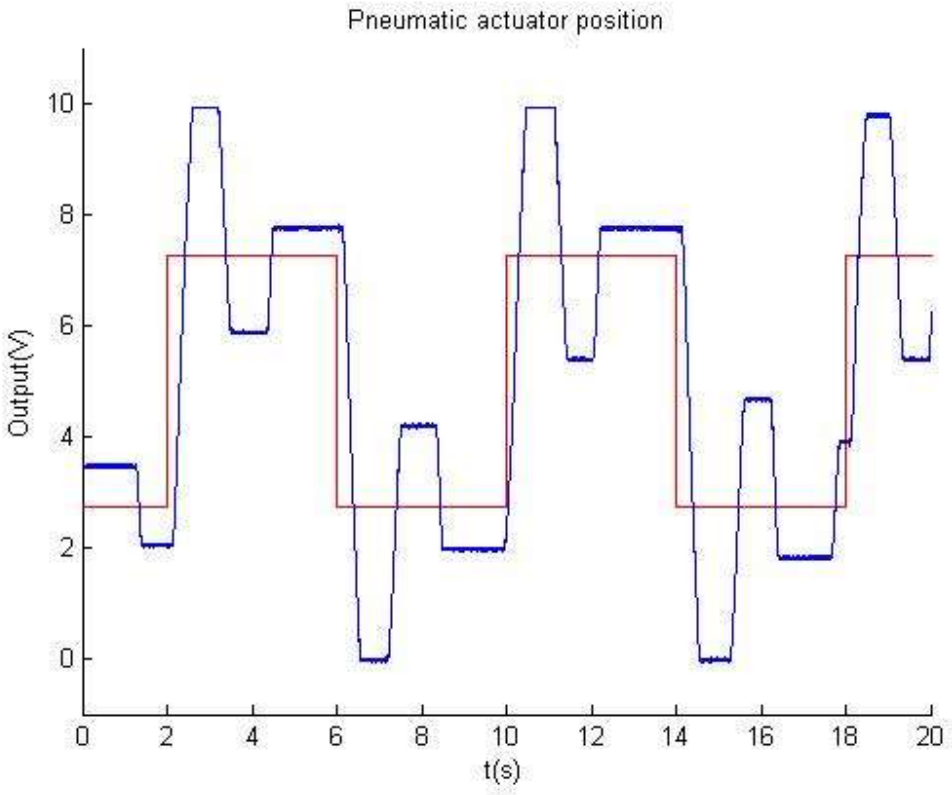


Figure 47: Pneumatic model test

6. Project related studies

Along this document it has been described the complete process followed to design, develop and manufacture both the laboratory practices board and the ESC study. In this chapter all the project will be summarize in terms of time spent, money inverted and social impact.

6.1 Project schedule

This whole process has been distributed into six main tasks described below.

1. Research and information gathering. This first task is crucial in any project because it gives the base knowledge required to fulfill the following steps.
2. Design and laboratory testing of initial prototypes.
3. Testing the Engine Control Hardware.
4. Design and manufacture of the board using Altium. This task includes the learning stage of the software Altium and the actual design and manufacture of the board.
5. Board assembly and final prototype testing.
6. Project documentation.

In order to illustrate the project schedule, a Gantt Chart, Figure 48, has been elaborated featuring the six tasks described above. That gives a better representation of the duration and start and finish dates of each task making it easier to comprehend.

Id	Nombre de tarea	Duración	Comienzo	Fin
1	Research and information gathering	85 días	lun 05/10/15	vie 29/01/16
2	Design and laboratory testing of initial	55 días	lun 07/12/15	vie 19/02/16
3	Testing the Engine Control System hardware	50 días	lun 14/12/15	vie 19/02/16
4	Design and manufacture of the board using Altium	60 días	lun 22/02/16	vie 13/05/16
5	Board assembly and final prototype testing	20 días	lun 16/05/16	vie 10/06/16
6	Project documentation	26 días	lun 16/05/16	lun 20/06/16

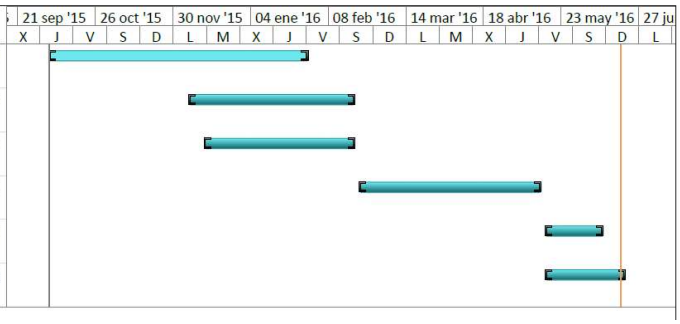


Figure 48: Gantt Chart of this project

6.2 Socioeconomic impact study

Automation at present has developed a few clear problems: the high cost reverberates in high prices of the final articles which can be easily surmountable with the globalization of the articles, and the rapid evolution of the technology which shortens the lifespan of any article due to obsolescence.

Extrapolating these problems to a social environment the consequences are clear and have a huge impact in modern society. Nowadays High cost technological products are directly related to social status and articles obsolescence creates the necessity of renovating them every year or even every few months.

This project was conceived to fight these main problems by creating a low cost device that can actually expand the lifespan of some articles. While it is true that is has not been design for particular use it still can affect a small part of the society through universities and work related spaces such as factories. It can also help as a small part in developing a new market in which consumers are educated in low cost articles with large lifespan.

Environmentally, lifespan increase leads to a diminish in the obsolete residues generated which is a major issue since electronics devices can be recycled but is an expensive long process and most people do not do it.

6.3 Budget study

In addition, a budget study is included taking into account every expense possible such as people involved in development, indirect costs, equipment used, ect.

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

Project Budget

1.- Author:		
	Rodrigo Barbosa Moreno	
2.- Department:		
	Automatic	engineering
3.- Project description:		
- Title	**Design and implementation of control hardware**	
- Duration (months)	**9**	
- Indirect cost taxes:		**20%**
4.- Project total budget (Euros):		
	Euros	35764
5.- Budgetary apportionment(indirect costs)		
PERSONAL		

Name and surname	N.I.F.	Rank	Time (Men / month) a)	Costs men/month	Coste (Euro)
Rodrigo Barbosa Moreno		Ing.Junior	10	2.450,00	24.500,00
Ramón Barber Castaño		Ing. Senior	2	2450,00	4.900,00
					0,00
					0,00
					0,00
		Men/month	**12**	**Total**	29.400,00

a) 1 men/month = 131,25 hours. Anual maximum 12 men/month (1575 horas)

Anual máximum for a PDI in Universidad Carlos III de Madrid is 8,8 men/ month (1.155 horas)

EQUIPMENT

Description	Costs (Euro)	% use during project	Time (month)	Depreciation period	d)
Laptop	840,00	100	10	48	175,00
Matlab license	6.000,00	20	2	120	20,00
Altium license	4.000,00	50	5	120	83,33
Welding station	300,00	10	1	48	0,00
Manufacturing components	150,00	100	10	12	125,00
					0,00
				Total	403,33

d) Depreciation formula:

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

OUTSOURCING TASKS

Description	Company	Imputable cost
Fabricacion de PCB en fresadora	UC3M	
	Total	0,00

COST RELATED TO THE PROJECT e)

Description	Company	Imputable cost
	Total	0,00

e) Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,…

6.- COSTS SUMMARY

Category	**Total Costs Budget**
Personal	29.400
Depreciation	403
Outsorucing tasks	0
Costs related to the project	0
Indirect costs	5.961
Total	35.764

7. Conclusions and future lines

7.1 Conclusions

As a result of this project a portable control system have been obtained, capable of communicating with a PC through an USB port and designed to work with two adjustable voltages ranges which gives a lot of flexibility to the system. Thanks to the limited size and the close box it can be easily transported and connected, adding the outstanding features of the STMF32Discovery makes the implementation in a host system really easy.

Along with the control hardware system some software applications were developed. Using Matlab as programing code, Simulink blocks specifically, a link between the microcontroller and the pc was implemented. These Simulink schematics are really basic and intuitive which gives the opportunity to work with them without needing extensive Matlab knowledge.

In addition to successfully build the control hardware, a good user level have been reach on the new software for printed circuits design, Altium Designer. All the new features that makes Altium different from others design software have been tested such as the 3D view generator which can help to visualize the final appearance of the PCB during the edition stage.

As for the ESC, is has been successfully controlled by Arduino but there is no room for them as drivers for automatic systems, mainly because of the accuracy of the control. The difference in cost do not compensate for the technical specifications.

In short, a control system has been designed, made, programmed and tested using both Altium and Matlab. The result fulfills without problems the requirements: limited size, easy integration in host systems, adaptability and low cost.

7.2 Future lines.

Given the results obtained the project objectives were fulfill but there is room for improvement in every aspect of the control system.

Regarding the conditioning circuit there are two major ways of improvement. The first one is reducing the size of the circuit. Using specific op-amps the number of resistors can diminish. Also adding a second voltage regulator for each way may help to standardize both paths since the resistors tolerance can be compensated and adjusted if every line has its own potentiometer. Additionally a high frequency filter could reduce some of the noise from all signals which can be very helpful since some systems are very old.

About Matlab and the Simulink diagrams the room of improvement is really big. In this project Matlab was used as a tool to test the control system but the possibilities that Simulink offers were never the objective. One possibility is to embed the control blocks into the microcontroller to eliminate all the time delays or further develop the PID designed in this project to achieve different states of control.

The ESC may not be suitable to control Robotic systems but it can be controlled by Arduino, adding a signal emitter and a receiver for the ESC and controlling an RC model such as a drone or a car should be possible with any microcontroller.

In terms of a pure generic control system if the gains of the conditioning circuit could be changed with an external potentiometer and the offset could be equally controlled, taking into account input and output impedances, a real time configurable conditioning circuit could be built.

GLOSARIO

Casos

REFERENCIAS

[1] Altium. Altium Designer .Available from: http://www.altium.com/altium-designer/overview / [17de junio de 2016]]

[2] Altium Designer footprint libraries. Available from: http://www.altium.com/community/libraries/altium-designer-libraries/altiumdesigner-footprint-libraries/en/altium-designer-footprint-libraries_home.cfm [17de junio de 2016]

[3] García, Vicente. Introducción al I2C en Arduino. Available from: http://www.hispavila.com/3ds/atmega/intro_i2c.html [17de junio de 2016]

[4] Wikipedia. ESC. Available from :https://en.wikipedia.org/wiki/Electronic_speed_control [17de junio de 2016]

[5] STMicroelectronics. TL084: Operational amplifiers. Available from: http://www.alldatasheet.com/datasheet-pdf/pdf/28776/TI/TL084.html [17de junio de 2016]

[6] STMicroelectronics.STMM32F4Discovery: microcontroller board. Available from http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/stm32f4discovery.html[17de junio de 2016]

[7] STMicroelectronics. LM317: Voltage regulator. Available from: http://www.alldatasheet.com/datasheet-pdf/pdf/22754/STMICROELECTRONICS/LM317T.html[17de junio de 2016]

[8] STMicroelectronics. LM337: Voltage regulator. Available from: http://www.alldatasheet.es/datasheet-pdf/pdf/22756/STMICROELECTRONICS/LM337.html[17de junio de 2016]

[9] “Implementación de Bloques de Simulink para Control Adaptativo de un Motor de Corriente Continua”, David Rodríguez Rosa, Proyecto fin de carrera, Universidad Carlos III de Madrid, Octubre 2012.

[10] “Ingeniería de Control. Modelado y control de sistemas dinámicos”. Luis Moreno, Santiago Garrido y Carlos Balaguer. Ariel Ciencia 2003.

[11] Matlab.Mathworks.http://www.mathworks.es/matlabcentral/fileexchange/?s_tid=gn_mlc_fx[17de junio de 2016]

[12]Simulink. Mathworks. http://www.mathworks.es/products/simulink/index.html[17de junio de 2016]

ANNEX

```
M48
;Layer_Color=9474304
;FILE_FORMAT=2:5
INCH,LZ
;TYPE=PLATED
T1F00S00C0.02756
T2F00S00C0.03346
T3F00S00C0.03543
T4F00S00C0.04331
%
T01
X0248Y0135
Y0145
Y0155
X0499Y0151
Y0141
Y0131
X03065Y0378
X02045Y0381
X01745
X03365Y0378
X0435
X0465
X0567
X0597
T02
X02725Y01555
X03125
X0518Y0151
X0558
X0656Y0277
X06095Y0297
Y0257
X0621Y0241
X0581
X05505Y02575
X0536Y0257
X05365Y0233
Y0193
X05055Y02585
X04855
Y02985
X05055
X0536Y0297
X05505Y02975
X0407Y02995
Y02595
X0359Y02485
X0342
X0359Y02885
X02885
Y02485
X0302
```

Annex 1: NC Drills File Part 1

```
X0302
X03005Y0236
X02605
Y0272
X0241
X0224
Y0312
X0241
X02605
X01565Y02905
X01695Y0259
X017Y0237
X013
X01295Y0259
X0131Y02895
Y03295
X01315Y03595
X01715
X01565Y03305
X024Y0328
X02495Y0327
Y03485
X024Y0368
X0298Y03605
X02895Y03485
Y0327
X03585Y03205
X0388Y03215
X0407
Y03615
X0388
X0381Y0383
X03815Y04005
Y04405
X0421Y0383
X04585Y03955
X03585Y03605
X0338
X05085Y03595
X05305Y03605
X05445Y036
Y032
X05305Y03205
X05085Y03195
X05155Y03945
X04985Y03955
X05555Y03945
X06175Y03665
Y03265
X06425Y03335
X0656Y03345
Y0317
Y03745
```

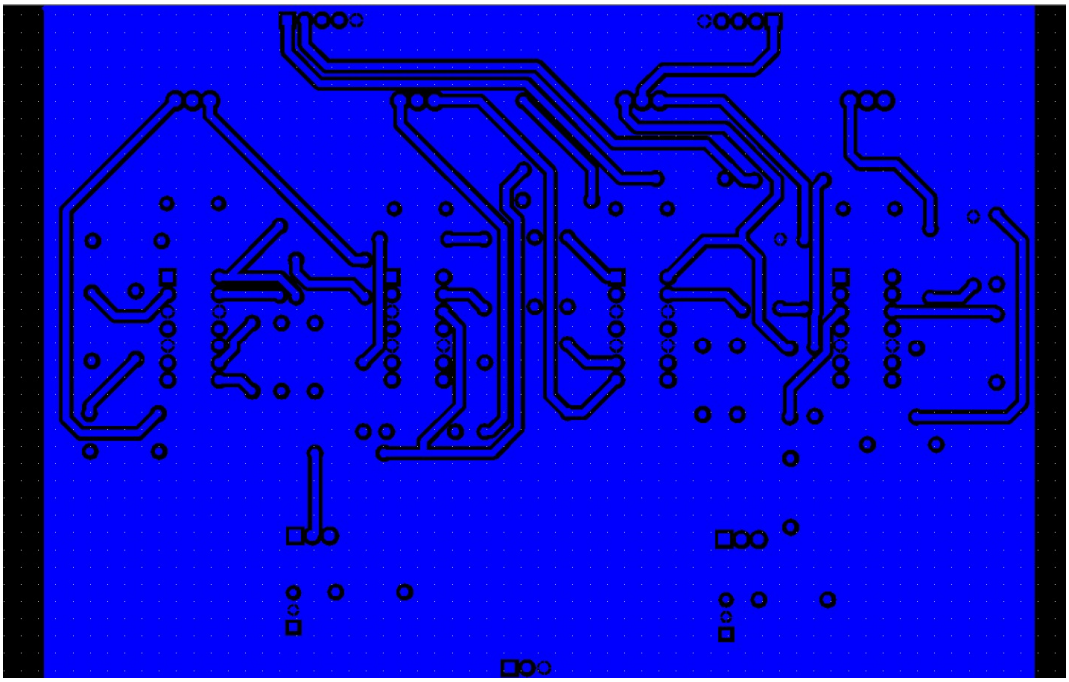
Annex 2: NC Drills File Part 2

```
Y03745
X06425Y03735
T03
X03735Y01115
X03835
X03935
X05655Y02785
Y02885
Y02985
Y03085
X05955
Y02985
Y02885
Y02785
X0465333
Y02885
Y02985
Y03085
X0435333
Y02985
Y02885
Y02785
X0335167
Y02885
Y02985
Y03085
X0305167
Y02985
Y02885
Y02785
X0205
Y02885
Y02985
Y03085
X0175
Y02985
Y02885
Y02785
Y03185
Y03285
Y03385
X0205
Y03285
Y03185
X0305167
Y03285
Y03385
X0335167
Y03285
Y03185
X0435333
Y03285
Y03385
```

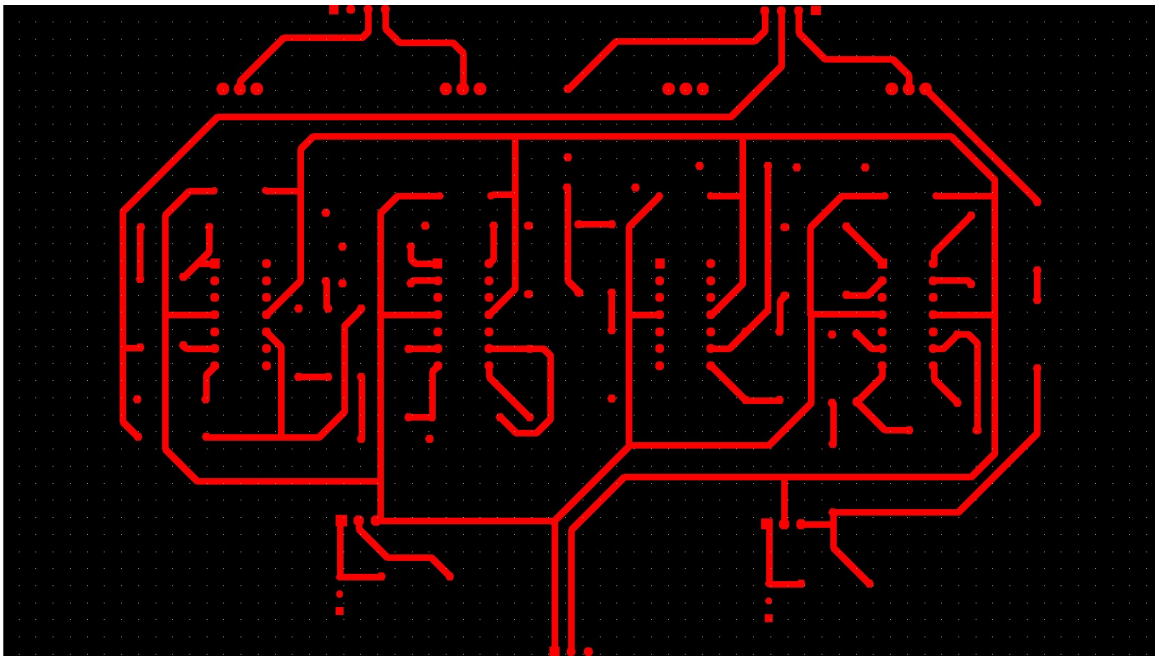
Annex 3: NC Drills File Part 3

```
Y03385
X0205
Y03285
Y03185
X0305167
Y03285
Y03385
X0335167
Y03285
Y03185
X0435333
Y03285
Y03385
X0465333
Y03285
Y03185
X05655
Y03285
Y03385
X05955
Y03285
Y03185
X05265Y04865
X05165
X05065
X04965
X04865
X02845Y0487
X02745
X02645
X02545
X02445
T04
X0498Y0186
X0508
X0518
X0269Y0188
X0259
X0249
X01995Y04405
X01895
X01795
X031
X032
X033
X04405
X04505
X04605
X0571
X0581
X0591
M30
```

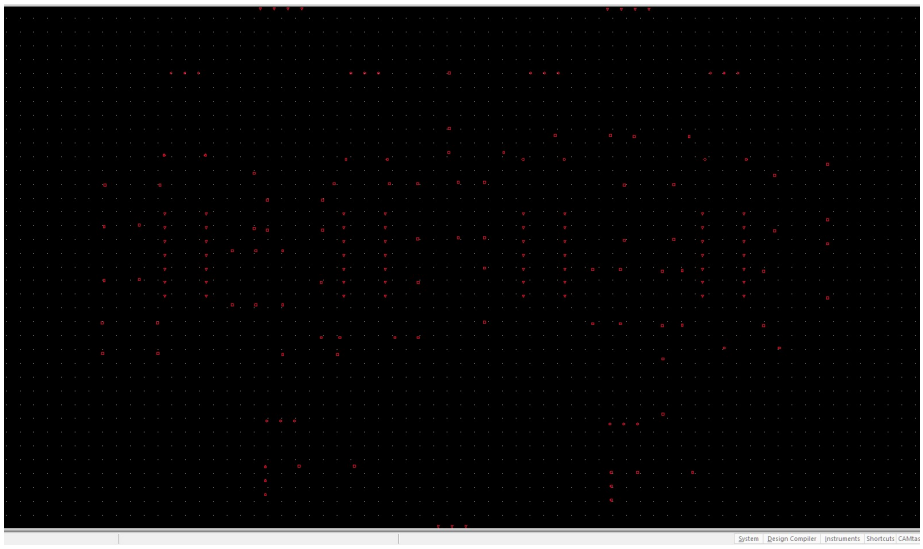
Annex 4:NC Drills File Part 4



Annex 5: GERBER file bot layer



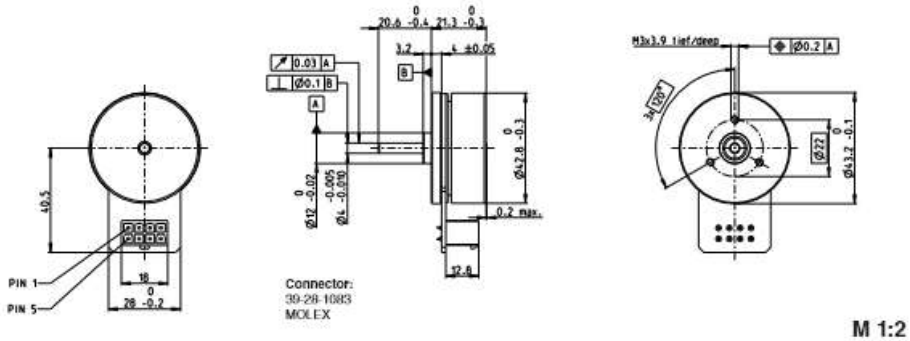
Annex 6: GERBER file top layer



Annex 7: GERBER file drills

maxon flat motor

EC 45 flat ∅45 mm, brushless, 50 Watt



Stock program
Standard program
Special program (on request)

		Order Number			
	with Hall sensors	339285	251601	339286	339287
Motor Data					
Values at nominal voltage					
1 Nominal voltage	V	18.0	24.0	24	36.0
2 No load speed	rpm	6710	6700	4710	3360
3 No load current	mA	294	201	125	46.9
4 Nominal speed	rpm	5250	5260	3460	2320
5 Nominal torque (max. continuous torque)	mNm	96.9	84.3	70.9	94.3
6 Nominal current (max. continuous current)	A	3.54	2.36	1.44	0.861
7 Stall torque	mNm	1100	822	411	488
8 Starting current	A	43.6	24.5	8.65	4.86
9 Max. efficiency	%	85	83	78	82
Characteristics					
10 Terminal resistance phase to phase	Ω	0.413	0.978	2.77	7.41
11 Terminal inductance phase to phase	mH	0.322	0.573	1.15	5.15
12 Torque constant	mNm / A	25.1	33.5	47.5	101
13 Speed constant	rpm / V	380	285	201	95.0
14 Speed / torque gradient	rpm / mNm	6.24	8.32	11.8	7.00
15 Mechanical time constant	ms	8.82	11.8	16.6	9.90
16 Rotor inertia	gcm²	135	135	135	135

Specifications

Thermal data
17 Thermal resistance housing ambient 4.25 K / W
18 Thermal resistance winding-housing 4.5 K / W
19 Thermal time constant winding 16.6 s
20 Thermal time constant motor 212 s
21 Ambient temperature -40 ... +100°C
22 Max. permissible winding temperature +125°C

Mechanical data (preloaded ball bearings)
23 Max. permissible speed 10000 rpm
24 Axial play at axial load < 4.0 N 0 mm
> 4.0 N 0.14 mm
25 Radial play preloaded
26 Max. axial load (dynamic) 3.8 N
27 Max. force for press fits (static) 50 N
(static, shaft supported) 1000 N
28 Max. radial loading, 7.5 mm from flange 21 N

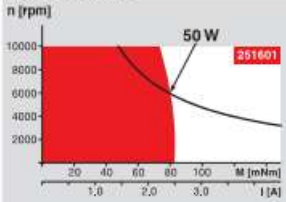
Other specifications
29 Number of pole pairs 8
30 Number of phases 3
31 Weight of motor 110 g

Values listed in the table are nominal.

Connection
Pin 1 Hall sensor 1*
Pin 2 Hall sensor 2*
Pin 3 4.5 ... 18 VDC
Pin 4 Motor winding 3
Pin 5 Hall sensor 3*
Pin 6 GND
Pin 7 Motor winding 1
Pin 8 Motor winding 2
*Internal pull-up (7 ... 13 kΩ) on pin 3
Wiring diagram for Hall sensors see p. 29

Cable
Connection cable Universal, L = 500 mm 339380
Connection cable to EPOS, L = 500 mm 354045

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
– Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

maxon Modular System Overview on page 16 - 21

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 237

Spur Gearhead
∅45 mm
0.5 - 2.0 Nm
Page 239

Recommended Electronics:

DECS 50/5	Page 289
DEC 24/1	290
DEC 50/5	291
DEC Module 50/5	291
DECV 50/5	297
EPOS2 Module 36/2	304
EPOS2 24/2	304
EPOS2 24/5	305
EPOS2 P 24/5	308
Notes	20

Option
With Cable and Connector
(Ambient temperature -20 ... +100°C)

May 2011 edition / subject to change

Annex 8: Maxom brushless motor datasheet